FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Univerzální laboratorní PLC úloha se zaměřením na vizualizaci průmyslových procesů.

Studijní program: B0714A270001 Mechatronika
Autor práce: **Jakub Fejfar**
Vedoucí práce: Ing. Martin Diblík, Ph.D.
Ústav mechatroniky a technické informatiky
Konzultant práce: Ing. Martin Vojíř
B&R Automatizace, spol. s r.o.

Liberec 2023

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Univerzální laboratorní PLC úloha se zaměřením na vizualizaci průmyslových procesů.

Jméno a příjmení:	**Jakub Fejfar**
Osobní číslo:	M19000074
Studijní program:	B0714A270001 Mechatronika
Zadávající katedra:	Ústav mechatroniky a technické informatiky
Akademický rok:	2021/2022

Zásady pro vypracování:

1. Prostudujte koncepci laboratorní úlohy s PLC technikou BR-Automation, seznamte se s mechanickým uspořádáním, elektroinstalací a jednotlivými automatizačními prvky.
2. Na základě zjištěných informací realizujte dvě identické úlohy. Sestavte mechanický rám, osaďte automatizační komponenty, realizujte jejich elektrické propojení.
3. Ve vývojovém prostředí BR Automation Studio navrhněte a realizujte program pro PLC automat. Software by měl demonstrovat základní funkcionalitu použitých automatizačních prvků.
4. V programu se zaměřte na názorné ukázky použití dostupných technologií pro vizualizaci průmyslových procesů a strojů.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30–40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] JOHN, Karl-Heinz a Michael TIEGELKAMP. IEC 61131-3: programming industrial automation systems: concepts and programming languages, requirements for programming systems, decision-making aids. Second edition. Berlin : New York: Springer, 2010. ISBN 978-3-642-12014-5

[2] BR-AUTOMATION. Training materials.

[3] MARTINÁSKOVÁ, Marie, Ladislav ŠMEJKAL, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. Řízení programovatelnými automaty. Praha: Vydavatelství ČVUT, 2004. ISBN 978-80-01-02925-1.

[4] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty II. Praha: ČVUT, Strojní fakulta, 2000. ISBN 978-80-01-02096-8.

[5] MARTINÁSKOVÁ, Marie, Ladislav ŠMEJKAL, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. Řízení programovatelnými automaty III: softwarové vybavení. Praha: Vydavatelství ČVUT, 2003. ISBN 978-80-01-02804-9.

Vedoucí práce:	Ing. Martin Diblík, Ph.D. Ústav mechatroniky a technické informatiky
Konzultant práce:	Ing. Martin Vojíř B&R Automatizace, spol. s r.o.
Datum zadání práce:	12. října 2021
Předpokládaný termín odevzdání:	22. května 2023

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 14. října 2021

Prohlášení

Prohlašuji, že svou bakalářská práce jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářská práce a konzultantem.

Jsem si vědom toho, že na mou bakalářská práce se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářská práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářská práce nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

25. 5. 2023 Jakub Fejfar

Univerzální laboratorní PLC úloha se zaměřením na vizualizaci průmyslových procesů

Abstrakt

Tato práce se zabývá sestavením stanice výukového modelu včetně veškeré elektroinstalace a montáží všech bezpečnostních prvků. Následně se pro tuto výukovou stanici vytvoří vizualizace v prostředí Automation Studio pomocí Visual Component 4.

Klíčová slova: Bezpečnostní prvky, PLC, Vizualizace, HMI

Universal laboratory PLC task with focusing on industrial visualization processes

Abstract

This work deals with the assembly of the training model station, including all wiring and installation of all safety features. Subsequently, a visualization is created for this training station in the Automation Studio environment using Visual Component 4.

Keywords: Safety features, PLC, Visualization, HMI

Poděkování

Rád bych poděkoval svému vedoucímu práce, kterým je Ing. Martin Diblík Ph.D. za poskytnuté rady, které byly pro mou práci velkým přínosem. Díky těmto radám jsem si uvědomil prostory pro zlepšení, které má práce nabízí.

Obsah

Seznam zkratek

TUL	Technická univerzita v Liberci
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
DC/AC	direct current (stejnosměrný proud) / alternating current (střídavý proud)
HMI	Human machine interface
PLC	Programmable Logic Controller
RFB	Remote framebuffer
px	Pixel

Úvod

Cílem této práce bylo vytvoření dvou laboratorních výukových stanic, na kterých bude možné provádět výuku programování PLC, tvorbu vizualizací, programování pohonů a tvorbu aplikací pro funkční bezpečnost zařízení.

V první části bylo zapotřebí sestavit rám z hliníkových profilů a následně jej osadit veškerými komponenty jako jako jsou pohony od výrobce B&R Industrial Automation, bezpečnostními prvky od společnosti SICK a veškerou elektronikou. Poté co byla jednotka sestavena, tak následovalo elektrické zapojení, které bylo realizováno dle přiložené elektrodokumentace.
Poté následovalo oživení celé stanice. Po oživení stanice bylo provedeno ověření správnosti zapojení všech bezpečnostních prvků a naprogramování jednoduchého demo algoritmu pro ověření funkčnosti pohonů ve vývojovém prostředí Automation Studio.

Jedním z požadavků této stanice je obsluze umožnit ovládaní.K tomuto účelu je zde umístěn HMI grafický panel, na kterém je zapotřebí vytvořit vizualizaci pro ovládání pohonů, kontrolu bezpečnostních prvků a dalších funkcí stanice. Tato vizualizace je hlavním cílem bakalářské práce.

Hlavním cílem vizualizace je především demonstrovat možnosti a typické způsoby použití vizualizačních prvků pomocí Visual Component 4. Je zde také krátce nastíněna tvorba vizualizace pomocí MappView, které Automation Studio také nabízí.

1 Požadavky na vizualizace

Za vizualizaci se považuje komunikace mezi člověkem a strojem, zařízením nebo systémem. Tato komunikace se také označuje jako HMI. Termín HMI se pak vztahuje prakticky na veškeré obrazovky, které slouží uživateli komunikovat se systémem (řídit ho a sledovat jeho chování).
U vizualizace máme různé možnosti jak zajistit tuto komunikaci mezi člověkem a strojem.
Pro komunikaci člověk-stroj nemusí být podmínkou přítomnost grafického panelu. Tato komunikace může být realizována i pomocí hardwarových komponent jako jsou světelné kontrolky, které zobrazují např. zda je rozvaděč nebo stroj pod napětím, zda je v poruše nebo mohou signalizovat i chod jednotlivých akčních členů.
Tyto světelné kontrolky zpracovávají pouze jednosměrnou komunikaci a to stroj->člověk. Proto se k těmto kontrolkám přidávají obyčejná hardwarová tlačítka například pro spuštění a zastavení akčního pohonu. Tyto tlačítka se často používají pro nouzové zastavení zařízení, která zůstanou po stisknutí zaaretovaná a pro jejich uvolnění je potřeba cílené povolení. Pro tuto situaci je použité i u této bakalářské práce.
Pokud je potřeba zobrazovat číselné údaje. Lze použít digitální panelové měřiče (zobrazovače). Tyto zobrazovače jsou nejčastěji tvořené LED nebo LCD displayem. Tyto displaye mohou zobrazovat například rychlost akčních členů, tlaku nebo teploty prostředí atd. Tyto hodnoty jsou ve většině případů zobrazovány pomocí analogových napěťových a proudových signálů.
Další možností toho rozhraní jsou alfanumerické displaye. Display je tvořen pomocí LED a lze na něm zobrazovat číslice, znaky i souvislý text (například alarmové hlášky). Pokud je tento text delší než počet diod displaye, tak je možné tento text zobrazovat na více řádcích, nebo případně tento text nebo hodnoty rolovat.
V dnešní době jsou ovšem nejrozšířenější grafické obrazovky s dotykovou vrstvou. Díky těmto obrazovkám je možné jejich použití pro obousměrnou komunikaci člověk-stroj. Tato je komunikace je hlavním tématem této bakalářské práce kde je realizována v prostředí Automation Studio pomocí Visual Compoment 4.

2 Koncepce laboratorní úlohy

Na těchto stanicích lze demonstrovat širokou škálu PLC úloh. Mezi tyto PLC úlohy patří ovládání servopohonů, tvorba vizualizace nebo programování bezpečnostních aplikací.

2.1 Řídící PLC

Základem řídícího systému je PLC typu Compact-S CPU řady X20 od výrobce B & R Automation. Toto PLC je osazeno I/O moduly. Konktrétně se jedná o moduly: Napájecí modul X20PS9600, modul digitálních vstupů X20DIF371, modul digitálních výstupů X20DOF322 a modul analogových vstupů X20AI4632-1. Všechny tyto karty jsou uloženy ve sběrnicovém modulu řídícího PLC.

2.2 Řízení pohybu

Všechny laboratorní stanice obsahují dva servomotory a jednou řídící jednotkou. Pohony se řadí mezi kompaktní servopohony 8LVA společnosti B & R Automation a řídící jednotku ACOPOS P3 v provedení SafeMotion. U těchto pohonů lze demonstrovat různé typy řízení. Jedním typem může být řízení každého pohonu zvlášť. To znamená, že se pro oba pohony nastavují parametry zvlášť a stejným způsobem se oba pohony i spouští. Další variantou řízení těchto pohonů je možnost vytvoření vazby mezi oběma pohony. V tomto stavu je možné pohony programovat tak aby se pohyb jednoho pohonu odvíjel v závislosti na pohonu druhém.

2.3 Bezpečnostní prvky

Stanice jsou vybaveny bezpečnostními komponety, kterými jsou: Dveřní zámek, dveřní snímač, optická clona a kombinovaný ovladač E-STOP a reset. Tyto prvky jsou zapojené do safety PLC, kde se vyhodnocují. Bez těchto komponent by nebylo možné realizovat bezpečnost zařízení.

2.4 Bezpečnostní PLC

Bezpečností systém je nedílnou součástí většiny zařízení. Všechny bezpečností normy a požadavky jsou definovány legislativou, která zajišťuje bezpečnost strojů. Bezpečností PLC je závislé na sygnálech z bezpečnostních prvků, které vyhodnocuje a zajišťuje bezpečnost celého zařízení.

2.5 Vizualizace

Dále je na stanici možné demonstrovat vizualizační prvky určené k ovládání nebo zobrazování aktuálních stavů laboratorní stanice. Tyto prvky jsou níže ukázány a popsány.

3 Hardware

Blokové schéma hardwaru laboratorní PLC úlohy se nachází v příloze.

3.1 Konstrukce

Konstrukce je složená z hliníkových profilů, které jsou spojeny pomocí konektoru universal. Tyto kulaté konektory se vkládají do hliníkových profilů kde se z druhé strany přitáhnou šroubem. Ostatní prvky se připevňují pomocí závitových destiček nebo závitových destiček pro následné vložení, oba typy jsou velikostí M3, M4 a M5.

Obrázek 3.1: Sestavené laboratorní stanice

3.2 Řídicí systém pracoviště

Řízení pohonů zajišťuje standardní programovatelný logický automat typu X20. Řídící systém je doplněn o následně zmíněné rozšiřující moduly.

Napájecí modul X20PS9600

Napájecí modul se používá společně s X20 Compact-S CPU. [1]

Modul digitálních vstupů X20DIF371

Tento modul obsahuje 16 digitálních vstupů pro jednovodičové připojení drátů.[2]
Pro toto zadání je modul nevyužitý.

Modul digitálních výstupů X20DOF322

Tento modul obsahuje 16 digitálních výstupů pro jednovodičové připojení drátů.[3]
V této práci jsou využité dva výstupy pro ovládání jednotlivých motorů.

Modul analogových vstupů X20AI4632-1

Tento modul obsahuje 4 šestnácti-bitové vstupy. Pomocí různého připojení na svorky si můžeme vybírat zda chceme proudový nebo napěťový signál.[4]
Modul je pro řešené zadání také nevyužitý.

3.3 Bezpečnostní řídicí systém

Bezpečnostní PLC X20SLX910

PLC modul je vybaven 20 bezpečnostními digitálními vstupy a 4 pulzními výstupy. Také je vybaven funkcí SafeLOGIC, která mu umožňuje bezpečně spouštět aplikace navržené v SafeDESIGNER. Modul lze použít v bezpečnostních aplikacích až do PL e nebo SIL 3.[5]
Na tuto kartu jsou připojeny všechny signály ze všech bezpečnostních prvků a vyhodnocují se v tomto PLC, protože je vybaveno safety digitálními vstupy. Nevýhodou je, že zvládá pouze jednoosý pohon s bezpečnostní funkcionalitou (SafeMotion). Jelikož je náš pohon dvouosý, tak pro jeho obsluhu využíváme X20SL8100. Mezi oběma PLC probíhá datová výměna.

Bezpečnostní modul X20SO6300

Tento modul obsahuje 6 bezpečných digitálních výstupů. Jmenovité výstupní proudy jsou 0,2 A. Modul lze použít pro ovládání pohonů v bezpečnostních aplikacích až do PL e nebo SIL 3. [6]

Bezpečnostní PLC X20SL8100

Tento modul je vybaven funkcí SafeLOGIC, která mu umožňuje bezpečně spouštět aplikace navržené v SafeDESIGNER. Modul lze použít v bezpečnostních aplikacích až do PL e nebo SIL 3. Modul neobsahuje žádné bezpečnostní vstupy. [7]

Kombinovaný ovladač E-STOP a Reset SICK ES11-SC4D8

E-STOP obsahuje dva nucené rozpínací kontakty, jeden rozpínací kontakt a tlačítko reset. Celkové rozměry (Š× V× H) jsou: 40mm × 123mm × 52mm. [8]

E-STOP je umístěn z přední strany modelu, vlevo od obrazovky. Slouží k okamžitému zastavení jednotky obsluhou v případě nebezpečí. Po zatlačení červeného tlačítka STOP je třeba ho odaretovat otočením a stisknutím tlačítka RESET pro opětovné uvedení jednotky do provozu.

Obrázek 3.2: E-STOP

Dveřní snímač SICK STR1-SAFMO3P5

Jedná se bezdotykové blokovací zařízení se dvěmi bezpečnými výstupy a jedním pomocným kontaktem. Snímač funguje na principu transpodéru. [9]
Jedna část je umístěna na vnitřní straně dveří a druhá na podstavě rámu, tak aby na sebe dolehly při zavření dveří.

Obrázek 3.3: Dveřní snímač

Optická závora SICK C4MT-01214ABB03BE0

Jedná se o prvek z kategorie bezpečnostních světelných závěsů, který má výšku chráněného pásu 120mm a dosah typicky 5 metrů Snímač má rozlišením 14 mm a odezvou <= 14ms. [10]

Obrázek 3.4: Optická závora

Závora je umístěná z přední strany v prostoru nad dveřmi a pod obrazovkou, kde je na každé straně jeden snímač.

Dveřní zámek SICK MLP1-SMMA0AC

Jedná se o blokovací zařízení s jištěním. Blokování se aktivuje napětím po zamčení a maximální zajišťovací síla je 550 N. [11]

Obrázek 3.5: Dveřní zámek

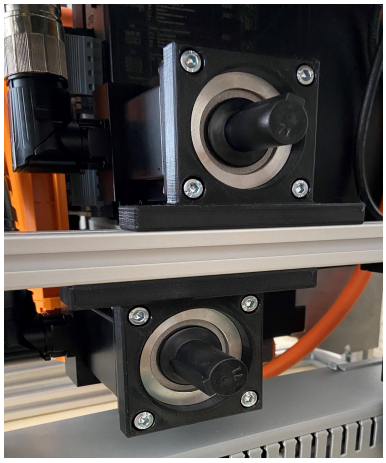
3.4 Pohony

3.4.1 Řídicí jednotka pohonů ACOPOS

Jedná se o jednotku pro řízení dvouosých pohonů. Pro řízení pohonů je zde použitý ACOPOS P3 typové číslo: 8EI4X5MWDS0.0200-1.
Input: 3× 200-230 VAC / 1× 110-230 VAC, 2,5kVA, 50/60 Hz
Output: 2× 0...230 VAC / 0 ...4,5 A, 0 ... 500 Hz [12]

3.4.2 Motory

Jednostka obsahuje dva motory od výrobce B & R automation, typové číslo: 8LVA22.B8030S100-0. Tyto pohony pjou připojeny k řídíci jednotce ACOPOS.



Obrázek 3.6: Pohony

3.5 Grafický operátorský panel

Pro vizualizaci je zde použitá obrazovka od společnosti B & R automation produktové číslo 6PPT30.0573-20W.

Obrázek 3.7: Grafický operátorský panel

Jedná se o Power Panel T30 o velikosti 5.7 palců, který je orientovaný na šířku. Má analogový odporový dotykový display o rozlišení 640 × 480 (VGA). Rozhraní: Dva Ethernety (10/100 Mbit/s - HUB) a dva USB 2.0. Klientský software obsahuje integrovanou servisní stránku, VNC klienta a vestavěný webový prohlížeč. [13]

3.6 Elektrické prvky

Jistící prvky

V projektu jsou použity dva jističe. Jeden který chrání zdroj 24V pro napájení veškeré elektroniky a druhý chrání ACOPOS, který řídí motory. Oba jističe jsou totožné a jedná se o typ LTE-B6-1 od výrobce OEZ. Jsou připevněny na DIN liště v pravé horní části stanice vedle zdroje pro 24V.

Zdroj 24V

Jako zdroj napájení pro PLC a ostatní elektroniku je zde použitý PS1050 od výrobce B&R Automation. Je umístěný na stejné DIN listě jako předešlé jističe. Výstupem je DC 24V/5A, který je přivedený na svorky odkud se rozvádí do potřebných zařízení.

Obrázek 3.8: Zdroj 24 V a jističe

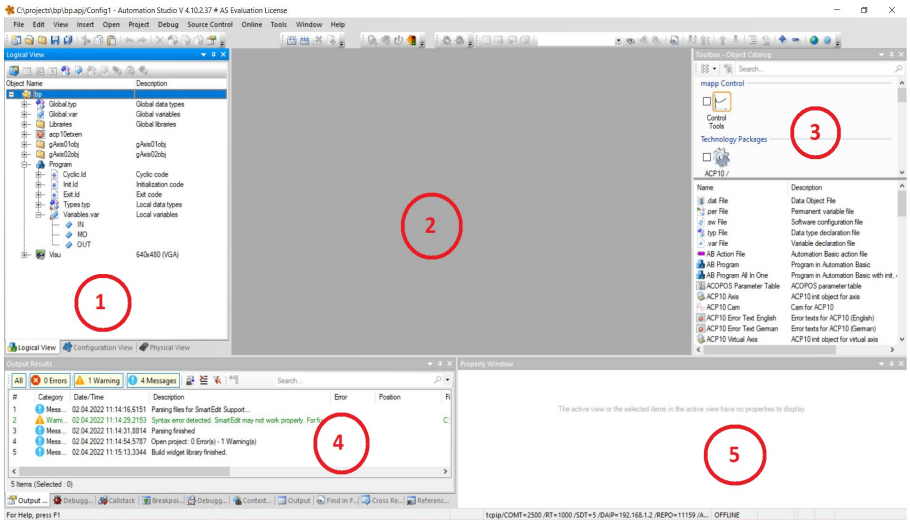
Svorky

Jsou zde použité dvě svorky pro napětí 24V. Jedna pro +24V (červená svorka) a druhá pro -24V (modrá svorka). Dále se zde nacházejí 4 svorky pro přívodní napětí ze sítě. Na šedou svorku je přivedená fáze, na modrou N vodič (tato svorka je zdvojena z důvodu potřeby více kontaktů) a poslední svorka je žluto-zelená a na ní je přiveden vodič PEN. Všechny svorky jsou opět umístěny na DIN liště.

3.7 Sestavená jednotka

Obrázek 3.9: Kompletně sestavená jednotka

4 Prostředí B & R Automation Studio



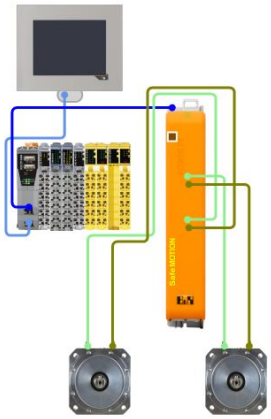
Obrázek 4.1: Náhled prostředí Automation Studia

Legenda

- 1: –Průzkumník projektu
- 2: –Pracovní prostředí
- 3: –Toolbox
- 4: –Debug okno
- 5: –Okno vlastností

4.1 Vytvoření HW a mapování vstupů

V prvním kroku jsem vkládal jednotlivé komponenty z toolboxu do systémového designeru a navzájem je propojil, čímž jsem vytvořil schéma stanice.

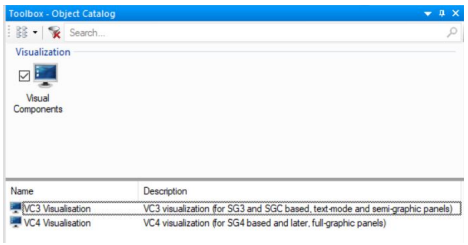


Obrázek 4.2: Schéma hardwaru

Poté jsem v programu ve `Varibles.var` založil vlastní datový typ `inputs` (označení `IN`) a `outputs` (označení `OUT`) a v `Types.typ` jsem vytvářel proměnné pro všechny signály z bezpečnostních prvků a přiřadil je buď do `IN` nebo do `OUT`.
Konečné mapování vstupů a výstupů jsem udělal v systémovém designeru, kde jsem si u každé karty na PLC rozkliknul `I/O mapping` a zde na příslušný kontakt přiřadil vhodnou proměnnou dle dostupného elektroprojektu.

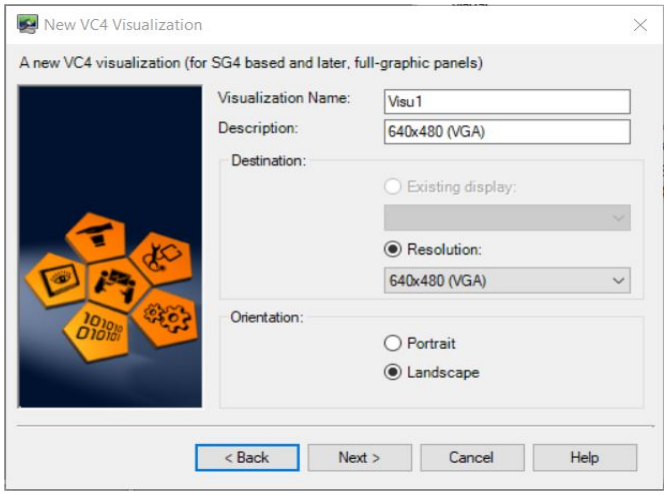
4.2 Založení vizualizace

V toolboxu jsem vyhledal Visual Components, Automation Studio našlo dvě možné visualizace: VC3 Visualisation a VC4 Visualisation. Pro použitou obrazovku vyhovuje pouze VC4 Visualisation.



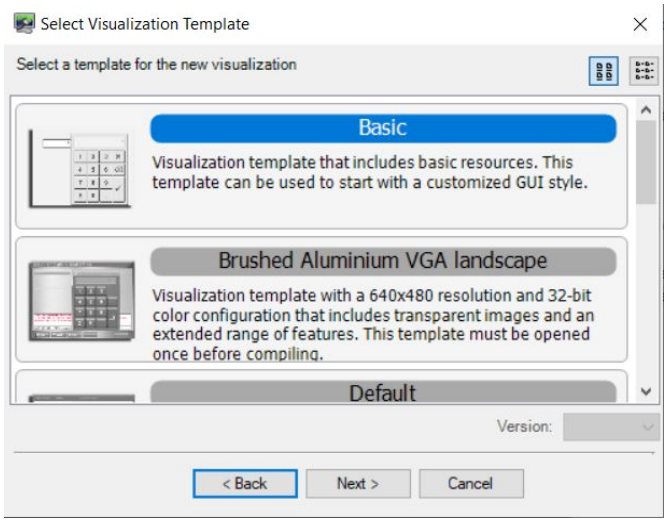
Obrázek 4.3: Schéma hardwaru

Po potvrzení VC4 Visualisation se zobrazí okno, kde je zapotřebí pojmenovat vizualizaci, vybrat dané rozlišení obrazovky (v tomto případě 640 × 480) a určit orientaci.



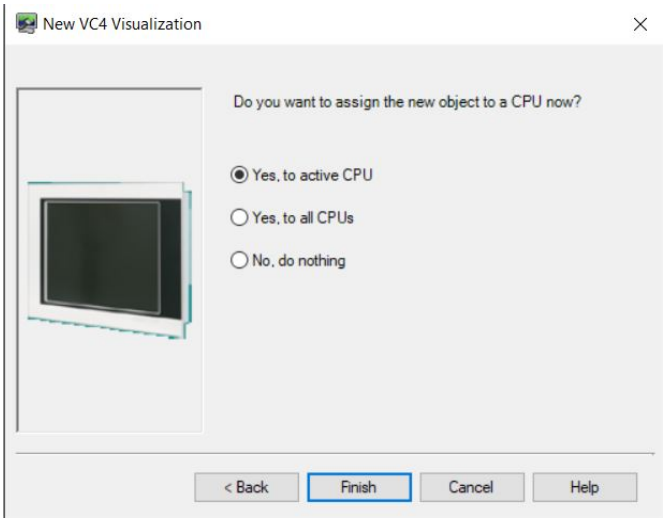
Obrázek 4.4: Nastavení vizualizace

V dalším kroku se provádí výběr z předpřipravených šablon pro vytvoření nové vizualizace. Pro tuto úlohu jsem zvolil šablonu `Basic`, která obsahuje základní zdroje.



Obrázek 4.5: Výběr šablony

V posledním kroku založení vizualizace se systém zeptá zda chceme hned přiřadit nový objekt k CPU, zde jsem zvolil první možnost (`Yes, to active CPU`). A kliknutím na `Finish` se vytvoří vizualizace.



Obrázek 4.6: Přiřazení k CPU

5 Vizualizace

Automation Studio nabízí dvě možnosti tvorby vizualizace. Mappview a pomocí Visual Component.

5.1 Mapp View

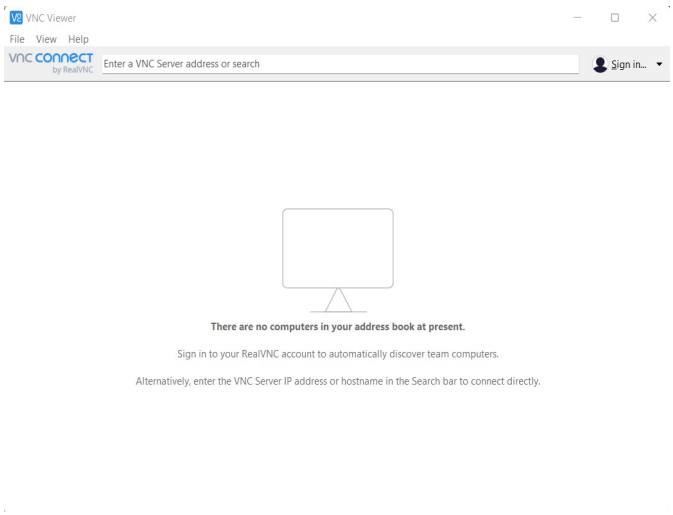
Mapp View je postaven na HTML5, CSS3 a JavaScriptu. Návrhář automatizace se může plně soustředit na klíčové kompetence. Stránky HMI jsou vytvořeny ve známém prostředí Automation Studio. Všechny funkce GUI jsou zapouzdřeny v modulárních ovládacích prvcích nazývaných widgety. Tyto widgety lze pohodlně přetáhnout na požadované místo a poté nakonfigurovat.

5.2 Visual Components

V závislosti na oblasti použití jsou ve Visual Components k dispozici dva různé editory. Podle používaných cílových systémů a odpovídající vizualizace je zapotřebí editor VC3 nebo VC4. V této práci se budeme zabývat VC4.

5.3 VNC klient

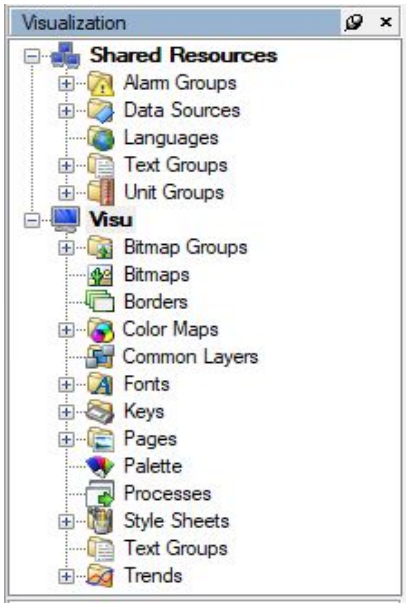
Virtual Network Computing, je grafický program, pomocí kterého se lze vzdáleně připojit ke grafickému uživatelskému rozhraní prostřednictvím lokální sítě nebo pomocí internetu. VNC pracuje na v režimu klient-server. To znamená, že server vytváří grafickou plochu a pomocí sítě komunikuje s klientem, který plochu zobrazuje. Ke komunikaci využívá protokol RFB, který se snaží minimalizovat objem přenášených dat mezi klientem a serverem. Pro připojení pomocí VNC je za potřebí tento zmíněný VNC klient (v této práci byl použit VNC Viewer) a adresa VNC serveru (127.0.0.1). V jiných případech se může používat i heslo. [**vnc**]



Obrázek 5.1: VNC klient

5.4 Vizualizační prvky

Následují text popisuje prvky (resources - lokální zdroje), které je možné použít při tvorbě vizualizace.



Obrázek 5.2: Visualization resources

`Bitmaps`

Do této záložky se vkládají a upravují všechny grafické prvky, které ve vizualizaci používáme. Pokud tam, žádný grafický prvek nepřidáme tak můžeme vybírat z před-

připravených prvků jako jsou bitmapy klávesnic nebo bitmapy pro tlačítka pomocí kterých se chceme pohybovat např.: v seznamech.

Borders

`Borders` se používají pro vytvoření ohraničení daného prvku (např. pro: `numeric`, `string` ...).

ColorMaps

V záložce `ColorMaps` se vytvářejí jednotlivé `ColorMapy`, na které se odkazují prvky, které mají měnit barvu v závislosti na určitých proměnných. Každá `ColorMapa` má definovaný index, přední a zadní barvu.

Common Layers

Zde se vytvářejí vrstvy, které se následně vkládají na jednotlivé stránky. Mohou sloužit k ukotvení menu na každé stránce nebo pro zobrazení nabídky nebo pro vytvoření vzhledu stránek, kterým přiřadíme danou vrstvu. V případě změny stačí upravovat pouze vrstvu v záložce `Common Layers`.

Fonts

`Fonts` složí pouze pro založení typu písma, který si pojmenujeme a nadefinujeme velikost písma, tučnost, kurzívu nebo typ písma.

Pages

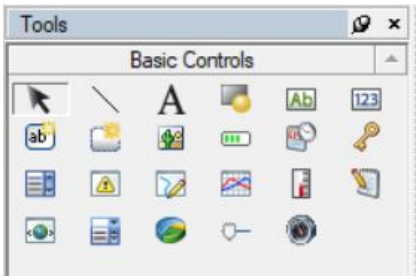
Jedná se o nejdůležitější prvek, jelikož se zde zakládají jednotlivé stránky vizualizace a také se zde s nimi pracuje.

Trends

V této záložce projektu se spravují všechna data trendů a jejich konfigurace, které lze přiřadit ovládacímu prvku trendu na příslušné stránce vizualizace.

5.5 Vizualizační komponenty

Jedná se o grafické ovládací prvky, které lze umisťovat na jednotlivé stránky vizualizace.



Obrázek 5.3: Nástroje

`Line`

Pomocí `line` lze vytvořit klasickou čáru, kde je definovaná stálá barva např.:pro oddělení určitých sekcí nebo může čára měnit barvu dle přiřazené `ColorMapy` a definováním `ColorDatapointu`, toto vše se nastavuje v záložce `Appearance` a `Format`. Čáru můžeme umisťovat ručně nebo v záložce `Position`, kde se nastavují hodnoty `StartX, StartY, EndX, EndY`.

`Text`

`Text` je klasické textové pole kde se nastavuje barva textu, průhlednost/neprůhlednost pozadí, font písma, zarovnání textu a také se zde lze nastavit text v angličtině a němčině (obsluha si může zvolit ve kterém jazyce chce HMI ovládat).

`Shape`

Pomocí `shape` se nejčastěji vytváří kontrolky. Opět musí být definovaná určitá `ColorMapa` pro přepínaní barev `shapu` a `ColorDatapoint` pro určení na jaký signál bude `Shape` reagovat.

`String`

Slouží buď pro zadávání textu obsluhou nebo pro zobrazování textu pomocí string znaků. V případě, že je `String` ve funkci input je potřeba připojit alfanumerický touchpad pro zadávání hodnot.

`Numeric`

Má obdobnou funkci jako `string` pouze s tím rozdílem, že zadává nebo zobrazuje číselné hodnoty typu integer nebo scaled. Pokud je `numeric` typu input je opět potřeba připojit nějaký numerický touchpad.

Button

Button je tlačítko, které se používá ve vizualizaci pro potvrzování nebo výběr. Vzhled tlačítka může být různý, např.: při stisknutí, uvolnění a nebo může zůstat zamáčknuté a dalším stiskem ho teprve uvolníme.

HotSpot

HotSpot má podobnou funkci jako Button s tím rozdílem, že ho nevidíme. Pomocí HotSpotu vytváříme plochu citlivou na dotek např.: když chceme kliknout na určitou část obrázku.

Bitmap

Prvek Bitmap slouží k vkládání obrázků.

BarGraph

BarGraph si můžeme představit jako ukazatel nabití baterie na mobilním telefonu. BarGraph může být buď spojitý nebo segmentovaný po daných hodnotách.

Trend

Pomocí této komponenty lze vkládat do stránek vizualizace grafy. Po vložení této komponety je za potřebí pouze vybrat nakonfigurovaná trend.

6 Tvorba vizualizace

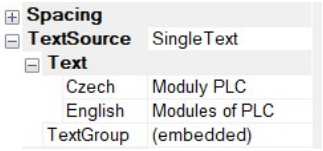
Následují kapitola stručně popíše postup tvorby vizualizace pro výše uvedené laboratorní pracoviště. Níže popisované činnosti představují základní výčet kroků, které je třeba provést, přičemž jsou uvedeny v doporučeném pořadí.

6.1 Jazyky

Vizualizace by měla být pro uživatele co nejpřívětivější, proto se většinou dělají v mnoha jazycích. U této vizualizace jsem se rozhodl udělat ji ve dvou jazycích a to v češtině a angličtině.
Editace jazyků se provází v `Shared Resources` v záložce `Languages`. Po vytvoření nové vizualizace jsou zde dva defaultní jazyky nastavené (Angličtina a Němčina). Nahrazení němčiny češtinou obsahuje pouze kliknutí na German (Němčina) a v pravém oknu vlastností vybereme z `listboxu Language Czech`.

Obrázek 6.1: `Languages`

Aby si uživatel mohl jazyk jednoduše přepínat, je potřeba u každého textu vizualizace ho definovat v obou jazycích. Všechny prvky obsahující text mají ve vlastnostech `Text Source`. Zde se nachází pro každý jazyk jedna kolonka.



Obrázek 6.2: `Text Source`

6.2 Fonty písma

Ve vizualizaci je potřeba si pro každý typ textu vytvořit `Font`, který se poté bude přiřazovat jednotlivým textům. Ve fontech se nastavuje typ písma, velikost a zda má být tučně nebo kurzívou. Všechny tyto parametry se nastavují pro všechny jazyky.

Je tedy možné, že jeden `Font` bude v každém jazyku jiný. Je to z důvodu délky slov v jednotlivých jazycích nebo kvůli speciálním znakům.

Index	Name	Description
0	DefaultFont	Default Font
1	Arial7px	
2	Arial9px	Font Arial with 9px
3	Arial9pxBold	Font Arial with 9px and bold
4	Arial9pxValue	Font Arial with 9px - used for Values
5	arial10px	
6	Arial10pxBold	Font Arial with 10px and bold
7	Arial12px	Font Arial with 12px and bold
8	arial12pxbold	
9	arial18px	
10	arial23pxbold	

Obrázek 6.3: Přehled fontů

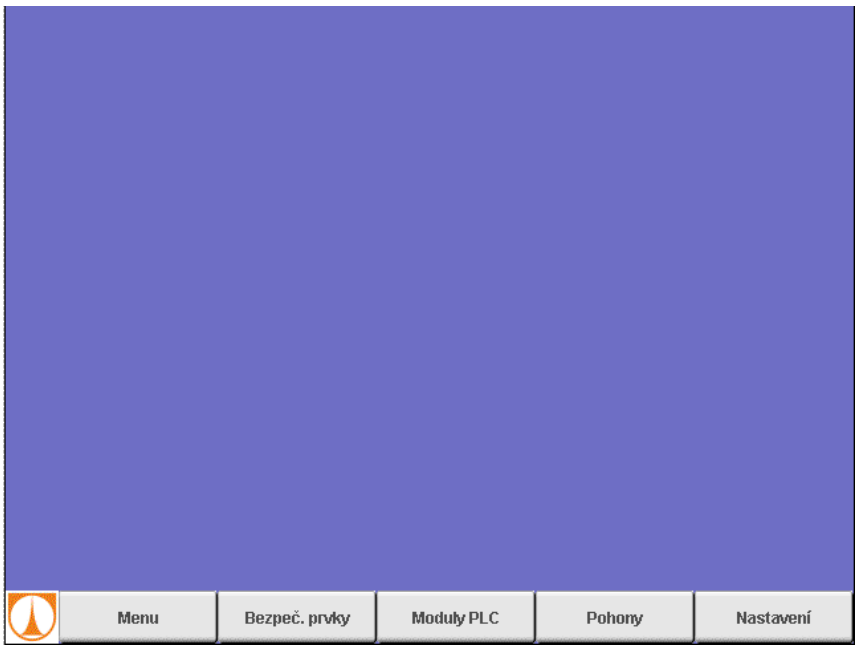
Language	Font Face	Size	Bold	Italic
Czech	Arial	12	False	False
English	Arial	12	False	False

Obrázek 6.4: Definování nového fontu

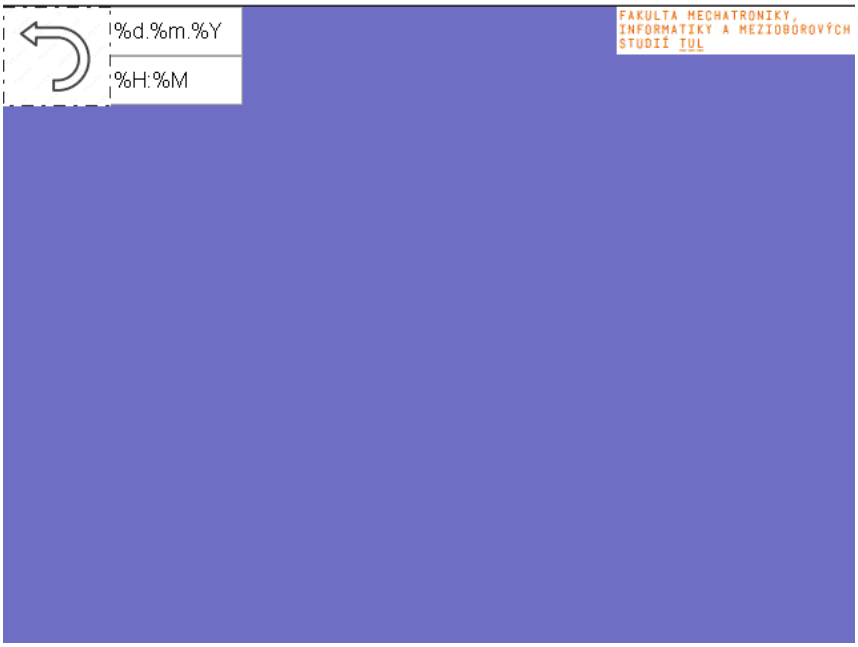
6.3 Vrstvy

V první části tvorby vizualizace jsem si musel rozmyslet rozmístění základních prvků jako je přepínání stran, krok zpět nebo zobrazení času, které se budou zobrazovat na většině stran. K tomuto jsem využil dvě vrstvy.
Jedna vrstva obsahuje ve spodní části tzv. lištu, na které je umístěno 5 tlačítek k přecházení mezi určitými stranami (Menu, Bezpečnostní prvky, PLC karty, Výběr motoru a Nastavení). Všechny tyto tlačítka mají nadefinovaný typ akce jako `ChangePage` a dále zvolenou příslušnou stranu, na kterou se chceme přepnout. Vrstva obsahuje v levém dolním rohu symbol FM TUL vložený pomocí `Bitmapy`, symbol zde slouží jen pro vylepšený vzhled.

Druhá vrstva slouží k zobrazení času a možnosti kroku zpět. V levém horním rohu se nachází `bitmapa` s motivem šipky zpět o rozměru 85×85 pixelů. Přes `bitmapu` je umístěný `HotSpot`, který má opět nadefinovaný typ akce na `ChangePage` a zde není nastavená určitá stránka, ale strana `Back`. Při kliknutí na `Bitmapu` zpět, respektive na neviditelný `Hotspot`, zobrazí se strana, na které jsme se nacházeli předtím.
Dále se na této straně nachází dvě vizualizační komponenty typu `DateTime`. Umístěny jsou nad sebou vlevo od `HotSpotu` pro krok zpět. V horním je zobrazován aktuální den, měsíc a rok. Ve spodním `DateTime` je aktuální hodina a minuta.



Obrázek 6.5: Vrstva Lišta

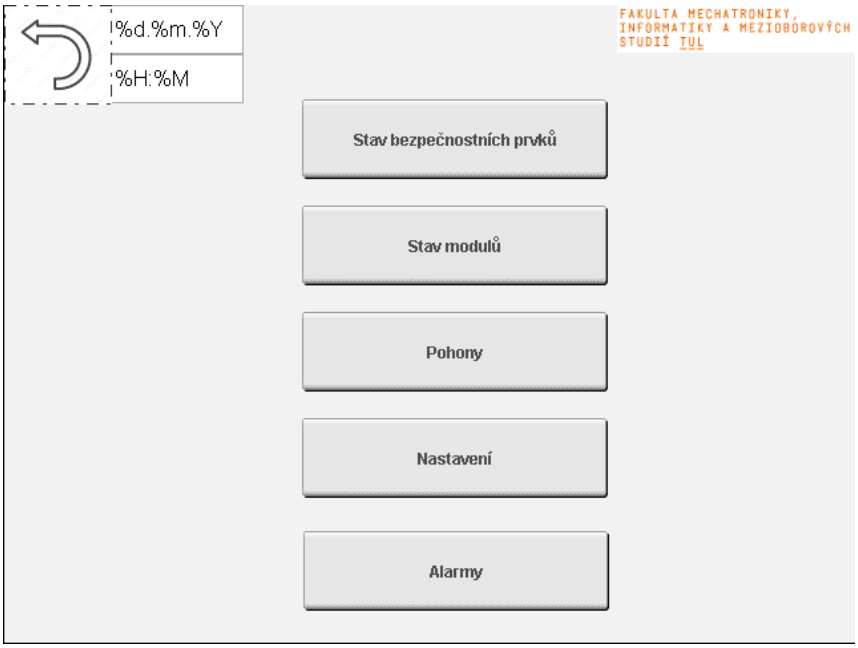


Obrázek 6.6: Vrstva Datetime

6.4 Stránky

6.4.1 Obrazovka Menu

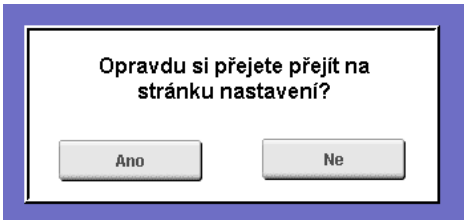
Stránka Menu je základní stranou, kterou uživatel vidí jako první po spuštění systému. Toto ošetření bylo nastavené v `Properties -> Runtime -> Description -> Default Page`. Na tuto stránku jsem vložil vrstvu nazvanou jako `dateTime`, kterou již mám vytvořenou. Vrstvy se na jednotlivé strany vkládají pravým kliknutím na příslušnou stránku (v tomto případě na stranu Menu) ve vizualizačních prvkách a poté zvolením možnosti `Add Layer`. Dále se pouze zvolí vrstva nebo vrstvy, které chceme na dané straně zobrazovat a potvrdíme `OK`.
Kromě vrstvy se zde nachází pět prvků typu `Button`, pomocí kterých se přepínáme na ostatní stránky. Tlačítka jsou definovaná stejným způsobem jako se definovaly tlačítka ve vrstvě Lišta. Těchto pět tlačítek odkazuje na strany: Stav bezpečnostních prvků, Stav modulů PLC, Výběr pohonu, Alarmy a Nastavení.



Obrázek 6.7: Menu

Na této obrazovce je použitý příklad zobrazení dialogového okna. Toto okno je využíváno při kliknutí na tlačítko nastavení. Poté, co uživatel klikne na tento `Button`, který není nastavený na `Change Page`, ale na `Set Datapoint`, tak se do proměnné `MessageBoxStatus` zapíše logická 0, pomocí které se zobrazuje dialogové okno.
Toto okno je vytvořené pomocí vrstvy v záložce `Common Layer` a pojmenovná jako `Potvrzeni`. Na této vrstvě se nachází pouze jeden `TextBox` a dva prvky typu `Button`. V `TextBoxu` je uvedený statický text, který se uživatele ptá zda chce opravdu přejít na stránku s nastavením a pomocí tlačítek (Ano/Ne) se uživatel rozhodne.

Oba `Buttony` jsou nastavené na `SetDatapoint`. V případě, že se uživatel rozhodne přejít na stránku s nastavením a potvrdí to kliknutím na `Ano`, zapíše se do



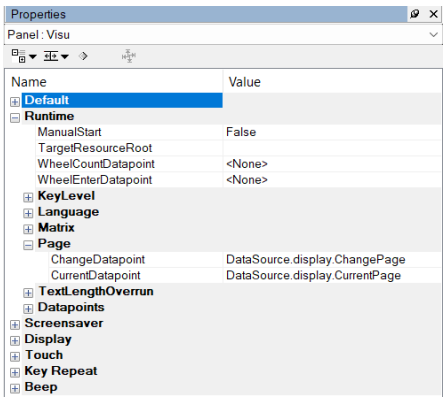
Obrázek 6.8: Vrstva Potrvrzeni

proměnné `CloseMessageBox` 2. Pokud se, ale rozhodne, že tam nechce přejít a zvolí tlačítko `Ne`, tak se do stejné proměnné zapíše 1. Toto se poté vyhodnocuje v jednoduchém programu.

```
IF ShowMessageBox = 1 THEN
    MessageBoxStatus.0 := 0;
    ShowMessageBox := 0;
END_IF;

IF CloseMessageBox = 1 THEN
    MessageBoxStatus.0 := 1;
    CloseMessageBox :=0;
END_IF;
IF CloseMessageBox = 2 THEN
    MessageBoxStatus.0 := 1;
    CloseMessageBox :=0;
    display.ChangePage :=5;
END_IF;
```

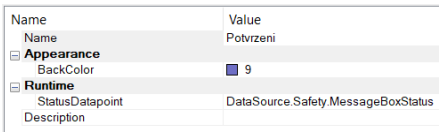
Obrázek 6.9: Vyhodnocení rozhodnutí

K úplnému fungování tohoto programu je potřeba nadefinovat dvě proměnné, jednu pro aktuální stránku vizualizace a druhou pro obrazovku na, kterou se má vizualizace přepnout. Toto se definuje v `Properties` celé vizualizace v záložce `Runtime Page`. Zde napojíme námi vytořené proměnné na tyto dva `DataPopinty`.



Obrázek 6.10: Vlastnosti vizualizace

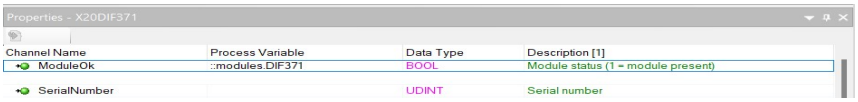
Dále je potřeba v `Properties` této vrstvy `Potvrzeni` nastavit proměnou `MessageBoxStatus` jako proměnnou, která definuje zobrazení.



Obrázek 6.11: Vlastnosti vrstvy Potrvrzeni

6.4.2 Obrazovka Moduly PLC

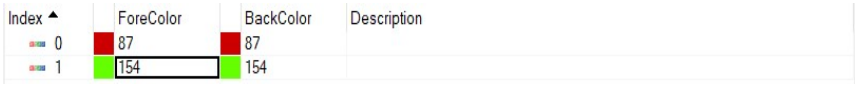
Tato stránka slouží k zobrazení zda jednotlivé moduly PLC pracují správně a komunikují spolu. Také je zde zobrazení teploty PLC a okolního prostředí. Na tuto stránku jsem již umístil obě vytvořené vrstvy pro jednoduší orientaci v programu. Jako první jsem na tuto stranu vložil `Bitmapu` PLC zařízení. `Bitmapu` jsem vytvořil screenshotem v `hardware designeru` a poté ji jako fotku upravil do rozměrů 357 * 177, aby se zachoval poměr stran a zároveň, aby se mi obrázek vešel na stránku. Poté jsem ho vložil do vizualizačních prvků `Bitmaps`, kde jsem musel zachovat jeho rozměry, které jsem upravoval v PC a pojmenoval ho plc_hw. Následně jsem pouze vložil na obrazovku komponentu `Bitmap` a nadefinoval ji `Bitmapu` plc_hw. Pokud bych chtěl upravit velikost `Bitmapy`, tak nelze ji jen intuitivně uchopit myší v rohu a roztáhnou, ale musí se celý proces s obrázkem opakovat od začátku. Kromě vrstev a `bitmapy` jsem přidal 6 `TextBoxů`, do kterých jsem vypsal názvy jednotlivých karet PLC a nadefinoval mu ohraničení `FlatBlack`. Tyto `TextBoxy` jsem umístil do řádku pod obrázek PLC a pomocí komponenty `Line` a `Shape` jsem udělal ukazatele na jednotlivé karty, aby byly zřejmé názvy karet. Následně jsem vložil 6 hranatých prvků `Shape`, které demonstrují LED diody. Aby byly `Shapy` správně nadefinované a signalizovaly správný stav karet, musel jsem nejdřív znovu vytvořit nový datový typ `modules`, kam jsem přidal 6 proměnných s názvy karet (`DOF 322, SLX910, SO6300, SL8100, AI4632_1 a DIF 371`). V dalším kroku jsem tyto proměnné mapoval na příslušné karty do `ChannelName Module OK`.



Obrázek 6.12: Mapování karet

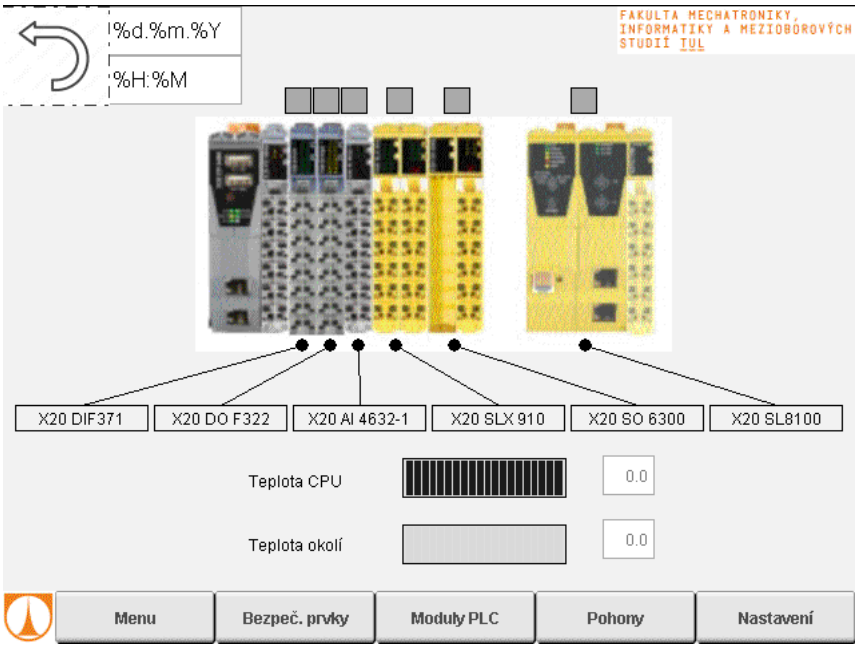
V tuto chvíli jsem měl fungující proměnné signalizující dva stavy: 0 modul není v pořádku a 1 modul je v pořádku. Tyto proměnné jsem také přiřadil ve vlastnostech jednotlivých `Shapů` do `ColorDatapoint`.
Posledním krokem bylo vytvoření `ColorMapy`, aby měl Shape definované barvy, mezi kterými má přepínat, když je proměnná `ColorDatapoint` v logické 0 a v logické 1. Toto se dělá v záložce `ColorMaps`. Zde jsem si přidal novou `ColorMapu` a pojmenoval si ji red_green. Tato `ColorMapa` musí obsahovat dva indexy, které představují hodnoty proměnné (zde tedy 0 a 1). Do indexu 0 jsem přiřadil do `ForeColor` (barva ohraničení) i do `BackColor` (barva výplně) hodnotu 87, což je hodnota pro sytě červenou barvu. Naopak jsem to udělal u indexu 1. Zde jsem do obou kolonek pro výběr barvy přiřadil hodnotu 154 (svítivě zelená). Podle těchto dvou barev je zřejmé

zda je všechno v pořádku či nikoliv. Nakonec jsem tuto `ColorMapu` nadefinoval ve vlastnostech každého Shapu v `ColorMap`.



Obrázek 6.13: Definování `ColorMapy`

Druhou částí této strany je zobrazování teploty okolí a teploty PLC. Toto je zde zajištěno pomocí komponent `TextBox`, `BarGraph` a `Numeric`. V `TextBoxech` jsou napsány teploty o které se jedná a pomocí `BarGraph` a `Numeric` zobrazuji teploty. `Bargraph` zobrazuje teplotu graficky jako škálu od 0 do 100 pomocí 20 segmentů s mezerami 1. `Numeric` zobrazuje hodnotu pomocí číslic, kde je teplota zaokrouhlená na jedno desetinné místo. Při nastavování proměnných jsem využil interních proměnných PLC, jelikož jsou zde již nastavené a není třeba žádné mapování apod..

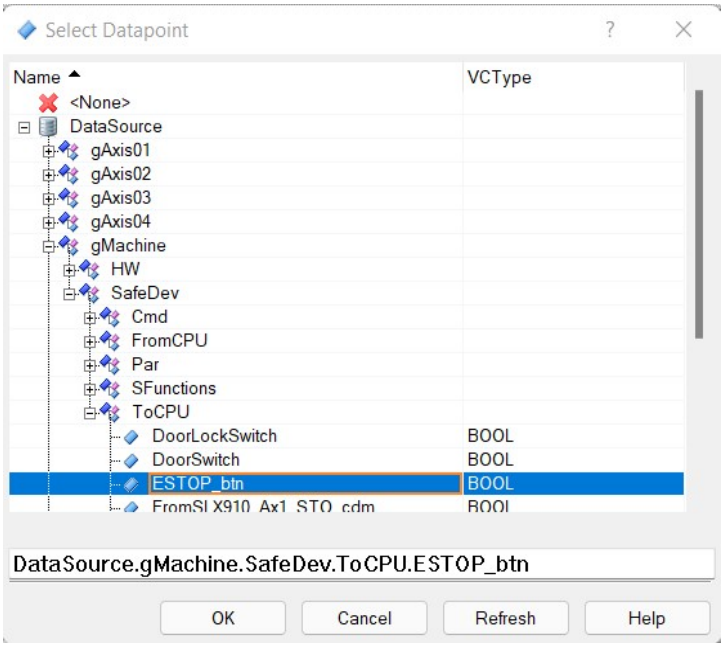


Obrázek 6.14: Moduly PLC

6.4.3 Obrazovka Stav bezpečnostních prvků

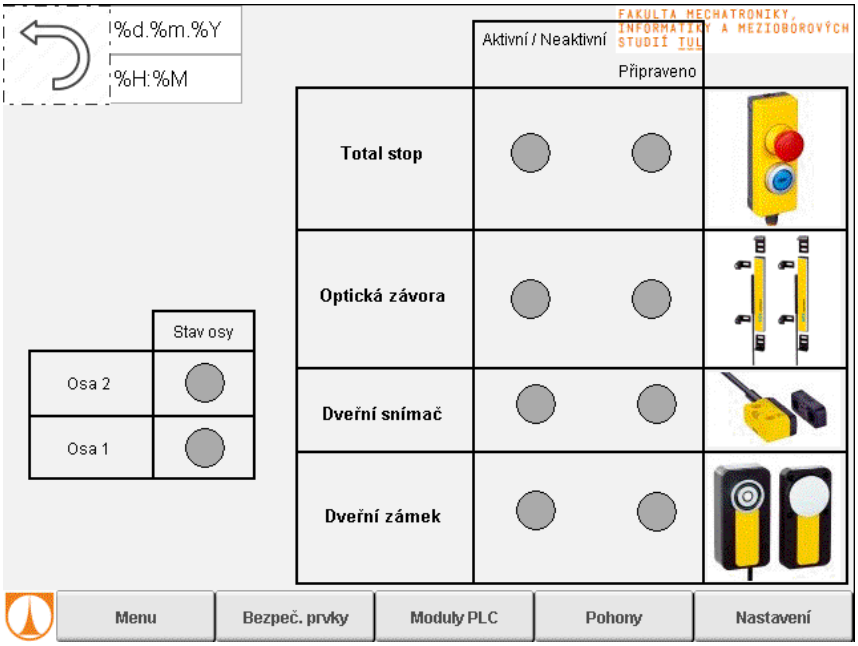
Zde jsou v tabulce zobrazeny stavy jednotlivých bezpečnostních prvků. Pro každý bezpečnostní prvek je zde napsaný jeho název v `textboxu`, jeho obrázek vložený pomocí `bitmapy` a dva `shapy`. V `textboxech` jsem zvolil tučný `font` písma Arial o velikosti 10px. Obrázky jsem upravil tak, aby každý obrázek měl na šířku 100px a poté je vložil jako `bitmapy`. Dále jsou zde pro každý bezpečnostní prvek dva kruhové `Shapy` o průměru 30px. První `Shape` slouží k zobrazení stavu snímače v logice sepnuto/rozepnuto a je použita Color Mapa red_green jako na předchozí

straně u zobrazování stavu karet na PLC. Tyto `Shapy` jsou namapovány k proměnným snímající oba safety kontakty každého bezpečnostního prvku `(DataSource -> gMachine -> SafeDev -> ToCPU -> zde vybrat odpovídající snímač)`.



Obrázek 6.15: `ColorDatapoint`

Pro druhý `Shape` jsem vytvořil další `ColorMapu` s názvem Green_green, která má opět dva indexy 0 a 1. Nula má nastavenou hodnotu `ForeColor` na 0 (obrys `Shapu` bude černý) a hodnotu `BackColor` na 156(výplň bude sytě zelená). Pro index 1 jsem nastavil hodnotu `ForeColor` opět na černou a hodnotu `Back Color` na hodnotu 154(svítivě zelená). Tento `Shape` indikuje zda bylo na stanici po nějakém zásahu stisknuto tlačítko resetu čí nikoliv `(DataSource -> gMachine -> SafeDev.-> SFunctions -> opět zvolen odpovídající kontakt)`. Stejně jsou nadefinované `Shapy` i v levé části obrazovky, které zobrazují připravenost pohonů k bezpečnému pohybu.
Tabulka je tvořena pouze pomocí čar o šířce 2px, které jsem umisťoval tak aby visuelně vytvořili tabulku pro lepší orientaci na obrazovce.



Obrázek 6.16: Bezpečnostní prvky

6.4.4 Obrazovka Výběru pohonu

Tato strana se zobrazí po kliknutí na tlačítko pohony. Ať už v menu nebo kdekoliv na liště. Strana slouží k vybrání pohonu, který chceme ovládat z HMI panelu. Jako na většině stran jsou zde umístěny obě vrstvy jako je Lišta a Datetime. Kromě vrstev je na této straně pouze obrázek celé stanice a `Listbox`, ve kterém se vybírá hodnota tzv. `request value`. Tato hodnota definuje v jakém bezpečnostním režimu má stanice fungovat.

Value	
Source	MultipleTexts
TextGroup	Requestvalue
TextIndexOffset	1 : 1
IndexDatapoint	DataSource.gMachine.SafeDev.Par.ModeRequestValue
MinIndex	<None>
MinDatapoint	<None>
MaxIndex	<None>
MaxDatapoint	<None>
OptionDatapoint	<None>
Input	True

Obrázek 6.17: `ListBox properties`

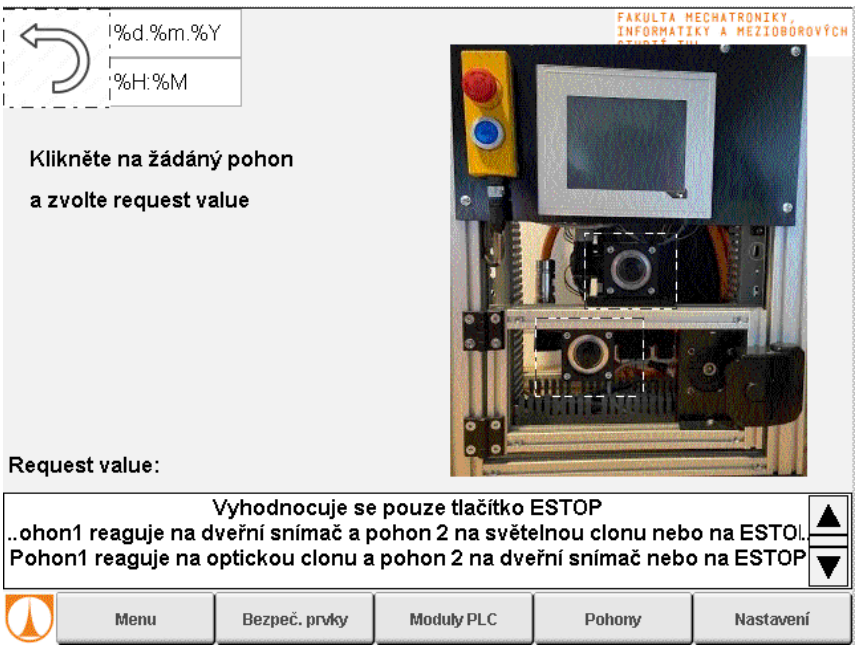
`Request value` má tři možné stavy

- 1 –Vyhodnocuje se pouze tlačítko ESTOP
- 2 –Axis01 se zablokuje, pokud DoorSwitch detekuje otevřené dveře NEBO je stisknuto tlačítko ESTOP a Axis02 se zablokuje, pokud LightGuard detekuje přerušení paprsku NEBO je stisknuto tlačítko ESTOP

- 4 - Axis01 se zablokuje, pokud DoorLock detekuje otevřené dveře NEBO je stisknuto tlačítko ESTOP a Axis02 se zablokuje, pokud LightGuard detekuje přerušení paprsku NEBO je stisknuto tlačítko ESTOP

V `listboxu` jsem nastavil tučný `font` písma a velikost 12 px, aby se pohodlněji klikalo na požadované hodnoty. Dále jsem definoval `IndexDatapoint` na `DataSource -> gMachine -> SafeDev -> Par -> ModeRequestValue`. A vytvořil novou `TextGroup` s názvem `Request value` a třemi indexy (1, 2, a 4) do kterých jsem přiřadil hodnoty 1, 2 a 4 pro všechny jazyky. Tuto `textgroup` jsem poté přiřadil v nastavení `listboxu`.

Poslední částí tvorby této strany bylo vytvoření dvou `Hotspotů`, které jsem umístil na obrázek přesně do míst kde se nacházejí pohony. `Hotspoty` jsem nadefinoval na `ChangePage` a přiřadil odpovídající stranu s pohonem.



Obrázek 6.18: Výběr pohonu

6.4.5 Základní obrazovka pro pohony 1 a 2

Strany pro pohon 1 a pohon 2 jsou naprosto totožné jen s tím rozdílem, že pohon 2 ovládá pohon, který je ve stanici umístěn výš a pohon 1 je pod ním. Tyto strany mají další sekce. Lze se přepnout na stranu kde se nastavují požadované hodnoty a provádí poříkazy, nebo na stranu s grafem vykreslující aktualní hodnoty.

Na základní obrazovce pohonů se nachází v `textboxu` tučný název pohonu o velikosti 23 px. Pro zobrazení aktuální pozice a rychlosti jsou využity dva prvky `numeric`. Proměnné s aktuální pozicí a rychlostí jsou pro každý motor v datovém typu `status`. Dále je zde ještě zobrazen stav daného pohonu. To je zajištěno pomocí dvou `textboxů`. První je použitý jako klasický `textbox`, ve kterém je text zapsán

staticky ve dvou jazycích a druhý `textbox` je proměnný a zobrazuje nám aktuální stav pohonu (Blokován, V pohybu, Error atd.). Aby se tento text měnil v závislosti na pohonu bylo potřeba vytvořit `TextGroup` s názvem `tg_Status`, který má 8 stavů. Pro těchto 8 stavů jsou indexy 100 až 107 a v každém je zapsán jeden stav v obou jazycích. Poté jsem v datovém typu `basic_axisState_typ`, ve kterém se nachází všechny možné stavy, založil novou proměnnou index typu INT. Do toto indexu přiřazuji v `programu.st` indexy 100 až 107 pomocí podmínky IF.

```
IF(Axis01_Control.AxisState.Disabled = TRUE) THEN
    Axis01_Control.AxisState.Index := 100;
END_IF
```

Obrázek 6.19: Podmínka IF

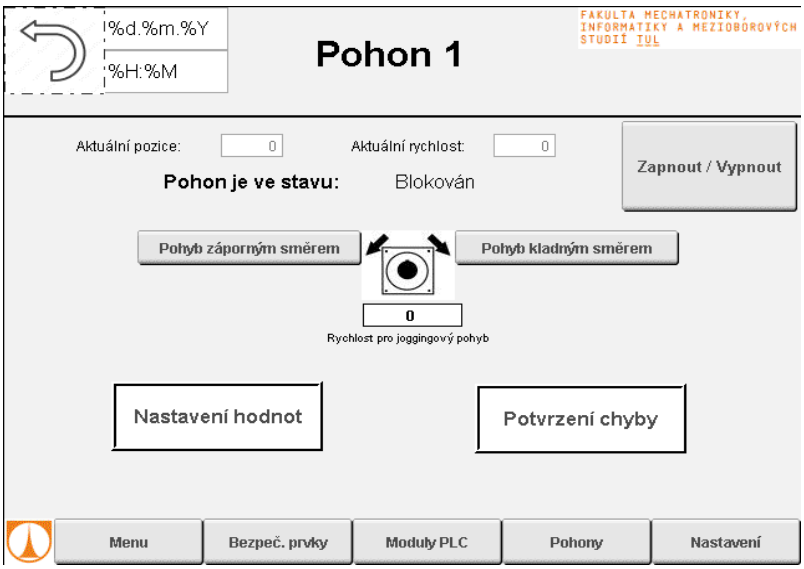
Hodnoty indexu - význam(česky/anglicky)

- 100 –Blokován/ Disabled
- 101 –V klidu/ Standstill
- 102 –Homování / Homing
- 103 –Zastavování / Stopping
- 104 –Diskrétní pohyb / Discrete Motion
- 105 –Nepřetržitý pohyb / Continuous Motion
- 106 –Synchronizovaný pohyb / Synchronized Motion
- 107 –Error / Error

Nakonec jsem nadefinoval `textbox` ve `value` na `Multiple Texts` a sem přiřadil vytvořenou `textgroup tg_Status` a `Datapoint Index` a ohraničil celou sekci pomocí `Line` o tloušťce 2 px.
Na této obrazovce se nachází i `Button` pomocí, kterého spouštíme vybraný pohon. Toho tlačítko je definováno jako `ToggleDatapoint`. Znamená to, že pokud tlačítko stiskneme a bude vizuelně zamáčknuté tak se do proměnné v `datapointu` bude přiřazovat 1 dokud ho nezmáčkneme podruhé a tím ho tzv. nevymáčkneme, poté bude `datapoint` 0.

Pod zobrazováním stavu pohonu a tlačítkem pro spouštění pohonu se nachází další část, která slouží k jogování pohonem buď v kladném nebo záporném směru. K tomu slouží pouze dvě tlačítka s definicí `MomentaryDatapoint`, dokud tlačítko držím, tak se do `datapointu` zapisuje 1, jakmile jej pustím objeví se tam 0. Mezi těmito dvěma tlačítky je vložen obrázek, který jsem vytvořil v malování jako symbol motoru a umístil k němu dvě šipky symbolizující otáčení motorem. Následně jsem obrázek vložil jako `bitmapu` o velikosti 75×60 px.

Posledními částí této strany jsou dvě tlačítka.
Pomocí prvního tlačítka se následnuje přepnutí na obrazovku, kde lze nastavovat jednotlivé parametry pro automatický pohyb pohonu. A pomocí druhého tlačítka se potvrzuje chyba daného pohonu.



Obrázek 6.20: Pohon 1 zvolený jazyk čeština

6.4.6 Obrazovka pro nastavování parametrů

První sekcí této strany je `Bitmapa` s grafem přes kterou je vytvořený `HotSpot` odkazující na další stranu, na které je umístěn graf se zobrazováním aktuálních hodnot.

Obrázek 6.21: HotSpot pro přechod do grafu

Na této straně se nastavují parametry pro veškeré pohyby motoru. Pro toto nastavování jsem vytvořil graf, na kterém je zřejmá náběžná a sestupná hrana. Horizontální osu grafu jsem popsal v (rychlost), vertikální osu jako s (dráha) a počátek souřadného systému 0. V počátku souřadného systému jsem ještě udělal pomocí kulatý `shape` o průměru 10 px a barvu výplně i obrysu jsem zvolil růžovou (index 69). Vedle tohoto `shapu` jsem vytvořil `numeric` pro input se stejnou barvou pozadí a `datapoint` nastavil k proměnné, kterou definujeme výchozí pozici `(Parameter -> Home position)`. Pod tento `numeric` jsem ještě vložil text `Home pozice` opět ve stejné barvě. Pomocí třech `line` jsem vytvořil průběh, na kterém je náběžná hrana jako zrychlení, konstantní rychlost a sestupná hrana jako brzdění. Náběžné hraně

jsem přiřadil oranžovou barvu (index 43) stejně jako textu a `numericu` pomocí kterého zadáváme zrychlení. Rychlost zadáváme na fialové konstantní hraně s indexem 13 do fialového `numericu` s popisem rychlost. Sestupná hrana má modrou barvu (169) a je definovaná stejně jako předchozí hrany. V místě kde se rychlost motoru dostává do nuly (když motor zastaví) je pomocí zelené úsečky protínající osu dráhy udělán bod, který vyznačuje vzdálenost od počátku. Pod tímto bodem je umístěn zelený `numeric` s popiskem vzdálenost, kde se nastavuje vzdálenost, kterou má motor vykonat. Dále se zde nachází `numeric` s popisem pro nastavení pozice, který v grafu nelze optimálně vyznačit, proto je umístěn mimo graf a obojí je znázorněno hnědou barvou s indexem 115.

Druhou částí této strany je označená nadpisem Příkazy. Jsou zde čtyři tlačítka typu `SetDatapoint` pomocí kterých volíme jakým pohybem se má motor pohybovat `(Move Absolute, Move Additive, Move Velocity)` a nebo ho může pomocí čtvrtého tlačítka uvést do `Home pozice`. Ve této druhé části je ještě nacházejí dvě tlačítka `MomentaryDataPoint`, které mám umožňují zastavit motor dvěma možnostmi. `Stop` nebo `Halt`.

Obrázek 6.22: Příkazy

U tlačítek `Move Additive` a `Move Absolute` je vytvořené zamykaní tlačítek. U těchto dvou pohybů je podmínka pro spuštění taková, že pohon musí být v pozici home aby bylo možné ho uvést do pohybu pomocí těchto dvou příkazů.

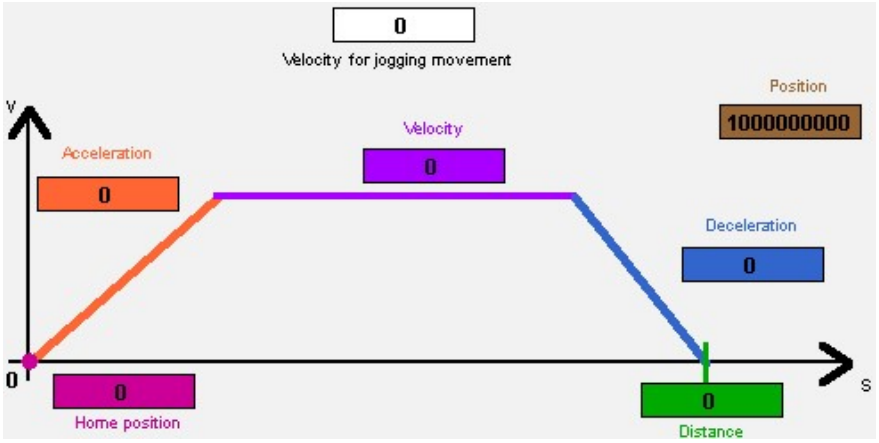
```
IF Axis01_Control.Status.DriveStatus.HomingOk = TRUE THEN

    status_lock_axis1.1 := 0;

ELSE
    status_lock_axis1.1 := 1;

END_IF;
```

Obrázek 6.23: Podmínka pro skrytí tlačítek

Poté co je podmínka vytvořená, napojíme proměnnou do které zapisujeme na druhý bit do datapointu v `Properties` u obou tlačítek.

Runtime	
ControllD	0
StatusDatapoint	DataSource.Motion.Axis01_Basic.status_lock_axis1
Locking	Never

Obrázek 6.24: Properties



Obrázek 6.25: Graf pro nastavování parametrů

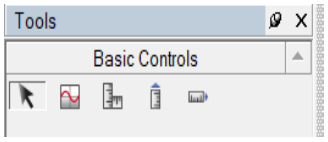
Vysvětlivky příkazů

Následující příkazy vyházejí z definice normy IEC 61131-3 a PLCopen funkčních bloků určených pro řízení pohybu(Motion Control).

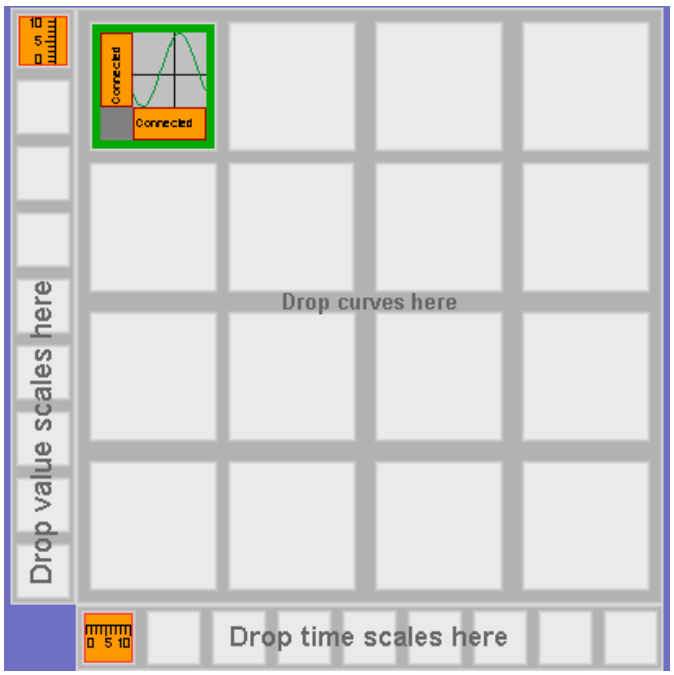
- `Move Additive` –Jedná se o pohyb zadané vzdálenosti
- `Move Absolute` –Jedná se o absolutní pohyb pomocí zadaných parametrů
- `Move Velocity` –Jedná se o řízený pohyb se zadanou rychlostí a zadaným směrem
- `Halt` –Jedná se o řízené zastavení pohybu. To však lze zrušit zahájením pohybu
- `Stop` –Tento funkční blok provede řízené zastavení pohybu a přepne osu do stavu zastavení
- `Home` –Výchozí pozice
- `Move jogging positive` –Tento funkční blok provádí pohyb s definovanými parametry v positivním směru, pokud zůstane nastaven odpovídající vstup funkčního bloku
- `Move jogging negative` –Tento funkční blok provádí pohyb s definovanými parametry v záporném směru, pokud zůstane nastaven odpovídající vstup funkčního bloku

6.4.7 Graf s aktuálními hodnotami

Pro vytvoření grafu je nutné ve `Visualization resources` v záložce `Trends` přidat Trend. Po přidání `Trendu` ho je možné otevřít a nadefinovat proměnné pro jednotlivé osy. Tyto osy se vkládají ze záložky `Tools`. Jsou zde měřítka pro osy X i Y a také křivka grafu.



Obrázek 6.26: Tools



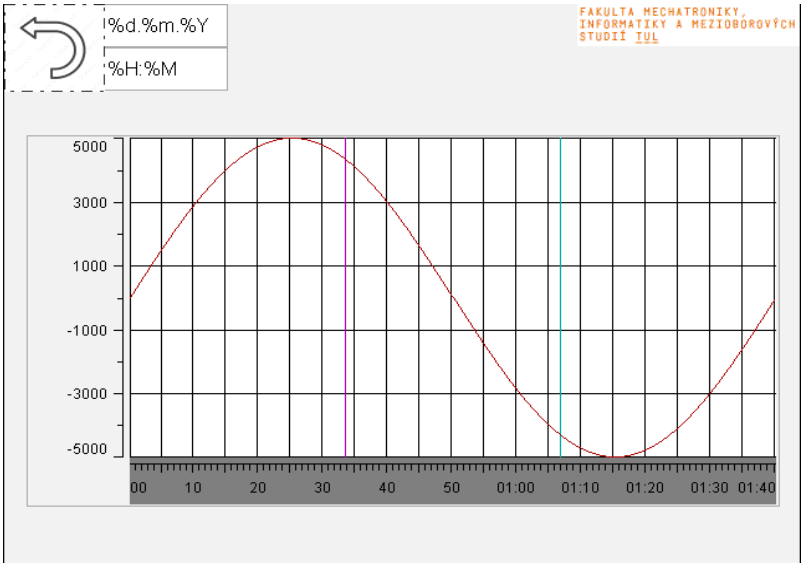
Obrázek 6.27: Definování křivek a os

Pro vytvoření grafu s vlastní proměnou je potřeba v záložce `Trend Data` přidat další data. Pro řešnou úlohu jsou zde potřeba tři `TrendData`. Rychlost pohonu 1, pohonu 2 a aktuální čas. U všech těchto dat je nutné mít nastavený typ na online a ne na frozen. Takto se bude graf měnit spojitě s časem. Tyto `TrendData` poté nadefinujeme u osy Y.

Index	Name	Type	Description
1	act_time	Online	
2	axis1_velocity	Online	
6	axis2_velocity	Online	

Obrázek 6.28: Definování křivek a os

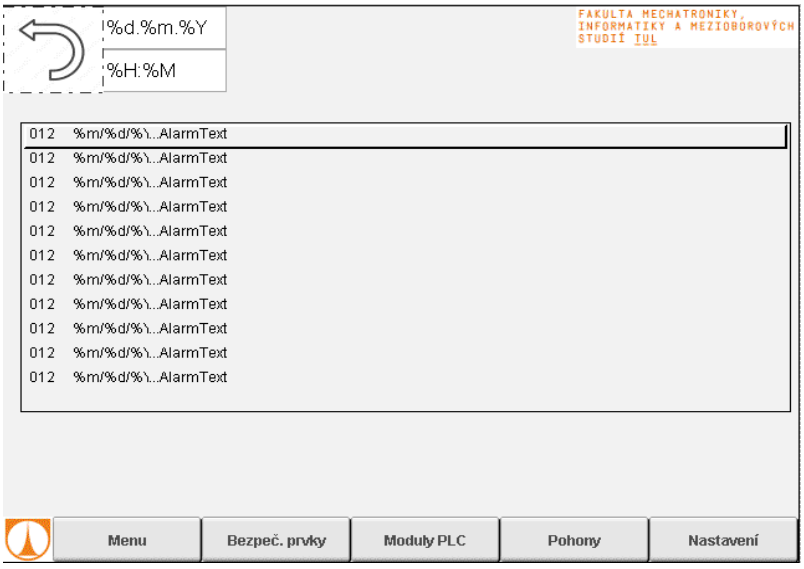
Poté co je všechno v `Visualization resources` v `Trends` nadefinováno zbývá už pouze vložit `Trend` ze záložky `Basic Control` na příslušnou stranu a zvolit nadefinovaná `TrendData`.



Obrázek 6.29: Graf s aktuálními hodnotami

6.4.8 Alarmy

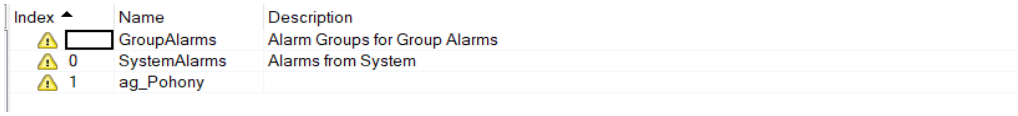
Na stranu s alarmy je možné přejít z výchozí obrazovky Menu, na které je `Button` s nápisem Alarmy. Po kliknutí na toto tlačítko se objeví následující obrazovka.



Obrázek 6.30: Obrazovka s alarmy

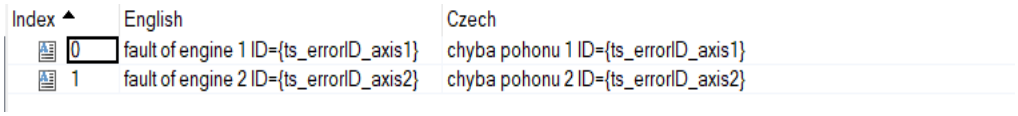
Pro správné zobrazení alarmů je potřeba postupovat následovně.
V dolní části stránky se nachází záložka `Text Snippets`. Po otevření této záložky

vytvoříme dvě Snippetky. Jednu pro čtení ID poruchy motoru 1 a druhou pro čtení ID poruchy druhého motoru. Tyto snippetky jsou pojmenovány `tsErrorIDaxis1` a `tsErrorIDaxis2`. U těchto snippetek nastavíme Datapointy na příslušné proměnné u obou pohonů, které zobrazují ID aktuálního erroru.
Dále v záložce `Shared resources Alarm Groups` vytvořit novou Skupinu (v tomto případě `agPohony`).



Obrázek 6.31: Alarm Groups

V této `Alarm Groups` jsou vytvořeny dva možné alarmy. Porucha pohonu 1 a 2. U obou pohonů nadefinujeme popis v Českém jazyce i Angličtině a pro zobrazení ID poruchy použijeme složené závorky a do nich uvedeme název textové snippetky, kterou jsme si vytvořili. Každý alarm, který je zde vytvořen tak má svůj vlastní index. Pro tuto úlohu stačí dva indexy a to 0 pro pohon 1 a index 1 pro pohon 2.



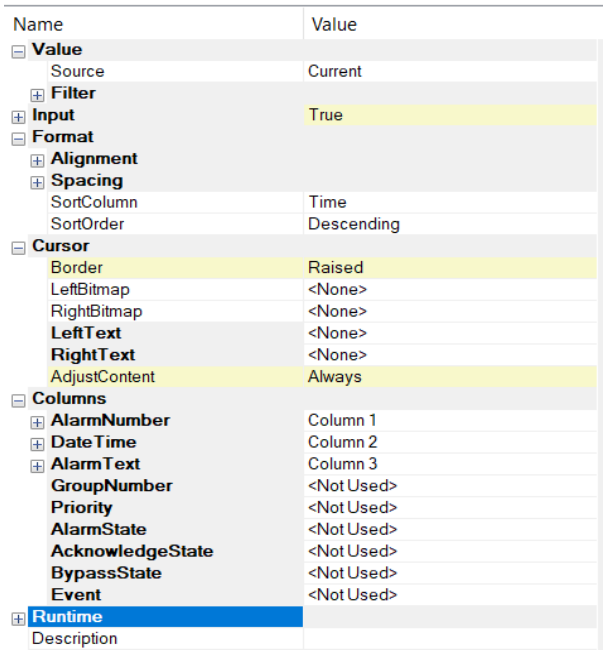
Obrázek 6.32: Text Snippets

V dalším kroku vytvořím pole `AlarmImage` o velikosti 2. Do kterého se zapisuje na nultý bit 0 nebo 1 v závislosti na tom jestli je pohon 1 v poruše či nikoliv. Stejným způsobem se zapisuje i do prvního bitu přítomnost poruchy pohonu 2.
Poté co je vytvořené pole, tak je možné do něj začít zapisovat. A to tak, že v programu obou pohonů přidám jednoduchou podmínku. Pokud je errorID daného pohonu aktivní přiřadí se na určitou pozici v poli 0 nebo 1.

```
IF Axis01_Control.Status.ErrorID <> 0 THEN
    AlarmImage[0] := 1;
ELSE
    AlarmImage[0] := 0;
END_IF;
```

Obrázek 6.33: IF

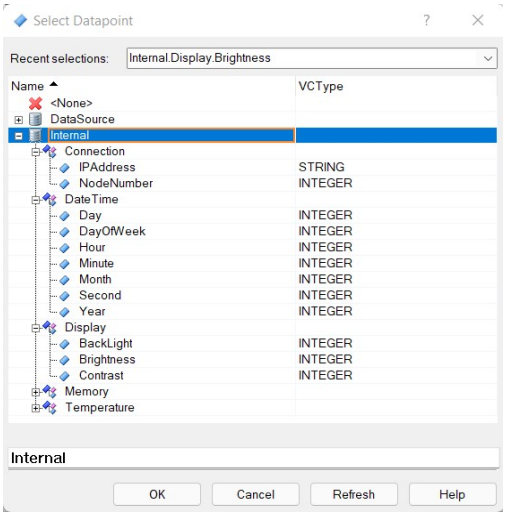
V tomto stádiu je již možné na dané obrazovce vizualizace vložit z `Basic Control` prvek `Alarm`. V properties tohoto vizualizačního prvku můžeme nasatvit zda se mají zobrazovat aktuální alarmy nebo historické alarmy. Toto se definuje v záložce `Value`, kde je potřeba nastavit `Current` pro zobrazování aktuálních alarmů.
Dále zde lze nastavit co se má u jednotlivých alarmů vypisovat. Pro běžné použití stačí datum poruchy, ID poruchy a písemný popis poruchy pro lepší přehlednost obsluhy.



Obrázek 6.34: Properties

6.4.9 Obrazovka Nastavení

Strana nastavení opět obsahuje obě vrstvy. Na této straně pracuje pouze s interními proměnnými jako je datum, čas, jas, kontrast a IP adresa.



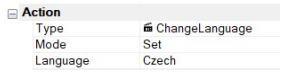
Obrázek 6.35: Interní proměnné

Pro nastavení datumu a času je využito 2 `textboxy` a 6 `numericů` (den, měsíc, rok, hodina, minuta a vteřina). V `textboxech` je napsaná posloupnost časových intervalů a za nimi jsou vždy tři okénka `numeric`. Každé okénko `numeric` je pro jiný časový interval a je na tento interval namapováno do proměnných `DateTime`. Dále je u každých z těchto intervalů nedefinována minimální a maximální hodnota, kterou lze do `numericů` zapsat. U vteřin a minut je nastaven rozsah 0 až 59, u hodin 0 až 23, u dní 0 až 31, u měsíců 0 až 12 a u let je minimální hodnota 0 a maximální jsem nedefinoval.

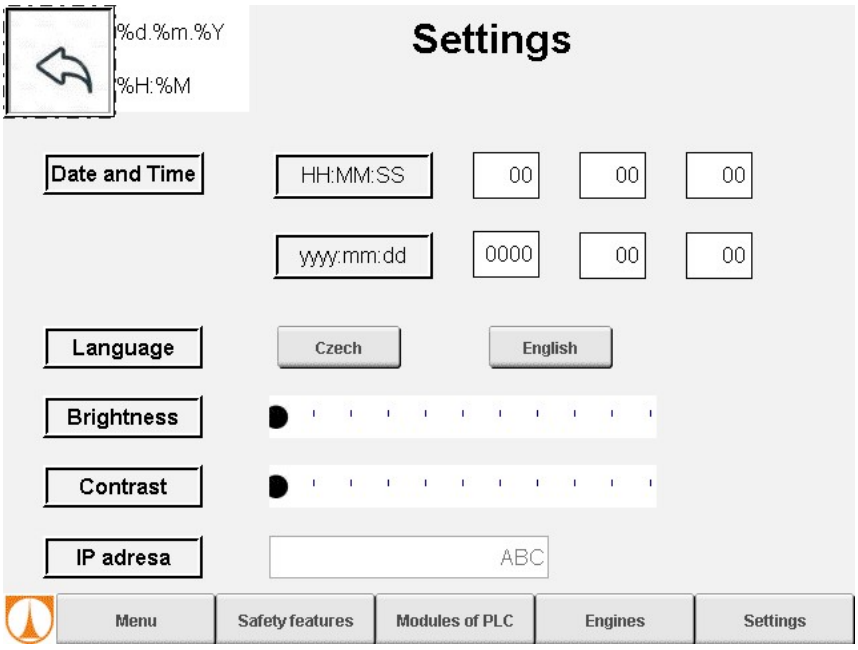
Nastavení jasu a kontrastu je zajištěno pomocí dvou `sliderů`, které mají `datapoint` opět v interních proměnných v `Display`. U `slideru` je potřeba také definovat i tzv. `thumb` (objekt se kterým bude posouváno po ose). Tento `thumb` jsem vytvořil v malování jako vyplněný kruh a vložil do `Automation Studia` do `Bitmap` o velikosti 20×20 px.

IP adresa je taktéž z interních proměnných a je zapsaná v `datapointu` prvku `String`.

Poslední možností, kterou lze v nastavení volit, je volba jazyka. Mezi jazyky se přepíná pomocí dvou tlačítek. Tyto tlačítka jsou nastaveny na `ChangeLanguage` a jedno má nastaveno češtinu a druhé angličtinu.



Obrázek 6.36: `ChangeLanguage`



Obrázek 6.37: Nastavení

7 Závěr

V první fázi této bakalářské práce jsem zkonstruoval dvě totožné stanice. Konstrukce začala sestavením vnější kostry tvořené hliníkovými profily a na tuto kostru jsem současně připojoval profily pro pozdější umístění elektronických komponent stanice.

Elektronické komponenty stanice obsahují pohony, bezpečností prvky, zdroj napětí, PLC, jističe a propojovací média. Jednotlivé komponenty jsem propojil dle elektrodokumentace, která byla poskytnuta vedoucím bakalářské práce. Další částí bylo propojení PLC s jednotlivými bezpečnostními prvky. Veškerá propojovací média jsem vedl v plastových lištách umístěných uvnitř stanice.

První problém nastal při oživení stanice, kdy jsem zjistil skutečnost, že použité bezpečnostní PLC X20SLX910 nedokáže ovládat dvouosý pohon a bylo zapotřebí přidat ještě jedno bezpečnostní PLC X20SL8100, pomocí kterého bylo již možné ovládat řídící jednotku APOPOS, a které také zároveň komunikuje s předešlým PLC. Pro toho řízení byly použity vzorové programy, které Automation Studio nabízí.

V této fázi jsem začal tvořit vizualizaci pomocí Visual Component 4, ve které jsem předvedl možnosti a typické způsoby použití jednotlivých vizualizačních prvků. Mezi tyto prvky patří jak základní vizualizační komponenty jako jsou `Buttony`, `Shape`, `HotSpot`, `ListBoxy`, `Bitmap`, `Numeric/String (input/output)`, `BarGraph` popis těchto komponentů je v kapitole 5.5 na straně 30 tak i pokročilejší, mezi které patří textové skupiny, alarmové hlášky v podkapitole 6.4.8 na straně 47, vyskakovací dialogová okna v podkapitole 6.4.1 na straně 35, grafy vykreslující aktuální hodnoty (v tomto případě rychlost nebo vzdálenost) v podkapitole 6.4.7 na straně 46, zablokování tlačítek proti stisknutí v podkapitole 6.4.6 na straně 44.

Tato práce shrnuje a představuje možnosti použití vizualizace VC4 a může sloužit jako manuál pro studenty, kteří s touto vizualizací začínají.

Nad rámec této bakalářské práce bylo vytvoření vizualizace pomocí MappView, které by bylo také skvělým námětem další práce.

Literatura

[1] AUTOMATION, BR. *X20PS9600*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20ps9600/.

[2] AUTOMATION, BR. *X20DIF371*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20dif371/.

[3] AUTOMATION, BR. *X20DOF322*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20dof322/.

[4] AUTOMATION, BR. *X20AI4632-1*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20ai4632-1/.

[5] AUTOMATION, BR. *X20SLX910*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20slx910/.

[6] AUTOMATION, BR. *X20SO6300*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20so6300/.

[7] AUTOMATION, BR. *X20SL8100*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/x20sl8100/.

[8] SICK. *Safety command devices ES11 / Emergency stop pushbutton / Complete device*. SICK, 2022. Dostupné také z: https://www.sick.com/us/en/safety-switches/safety-command-devices/es11/es11-sc4d8/p/p339946.

[9] SICK. *Non-contact safety switches STR1 / Transponder safety switch / Sensor with actuator*. SICK, 2022. Dostupné také z: https://www.sick.com/ch/en/safety-switches/non-contact-safety-switches/str1/str1-safm03p5/p/p416381.

[10] SICK. *Bezpečnostní světelné závěsy miniTwin / 1 Twin-Stick*. SICK, 2022. Dostupné také z: https://www.sick.com/cz/cs/bezpecnostni-svetelne-zavesy/bezpecnostni-svetelne-zavesy/minitwin/c4mt-01214abb03be0/p/p123669.

[11] SICK. *Blokovací zařízení s jištěním MLP1*. SICK, 2022. Dostupné také z: https://www.sick.com/cz/cs/blokovaci-zarizeni/blokovaci-zarizeni-s-jistenim/mlp1/mlp1-smma0ac/p/p494248.

[12] AUTOMATION, BR. *8EI4X5MWDS0.XXXX-1*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/products/motion-control/acopos-p3/servo-drives-safemotion-two-axis-modules/wall-mounting/8ei4x5mwds0xxxx-1/.

[13] AUTOMATION, BR. *6PPT30.0573-20W*. BR automation, 2022. Dostupné také z: https://www.br-automation.com/cs/produkty/6ppt300573-20w/.

A Přílohy

- Složka `priloha_bp_fejfar` tato složka obsahuje kompletní program, elektrodokumentaci k zapojení stanice a blokové schéma hardwaru