



Grafické rozhraní servisního technika pro ovládání skladovacího robotu

Bakalářská práce

Studijní program:

B2612 Elektrotechnika a informatika

Studijní obor:

Elektronické informační a řídicí systémy

Autor práce:

Tomáš Kubíček

Vedoucí práce:

Ing. Leoš Beran, Ph.D.

Ústav mechatroniky a technické informatiky

Konzultant práce:

Eva Doubková

Systematic a.s.





Zadání bakalářské práce

Grafické rozhraní servisního technika pro ovládání skladovacího robotu

Jméno a příjmení: **Tomáš Kubíček**
Osobní číslo: M17000044
Studijní program: B2612 Elektrotechnika a informatika
Studijní obor: Elektronické informační a řídicí systémy
Zadávací katedra: Ústav mechatroniky a technické informatiky
Akademický rok: **2019/2020**

Zásady pro vypracování:

1. Seznamte se s řešením skladového systému SysLogeum 3000.
2. Navrhněte vizuální podobu ovládání skladovacího robotu na základě zkušeností provozovatele systému.
3. Navržené rozhraní programově propojte s řídicím softwarem robotu.
4. Finální návrh a realizaci otestujte ve spolupráci s provozovatelem systému.
5. Napište návod na ovládání robotu pomocí vytvořené vizualizace.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30–40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. 1. díl. Praha: BEN, 1999. ISBN 80-8605- 58-9.
- [2] JOHN, Kharl-Heinz; TIEGELKAMP, Michael. IEC 61131-3 Programming Industrial Automation Systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision – Making Aids. 2nd.
- [3] Download SinuTrain for SINUMERIK Operate V4.8 – Basic. Siemens [online]. [cit. 2018-10-09]. Dostupné z: https://www.industry.siemens.com/topics/global/en/cnc4you/cnc_downloads/sinutrain_downloads/Pages/download-sinutrain-for-sinumerik-operate-v48-bas
- [4] REISING, D. V. (2013). *Effective console operator HMI design: ASM consortium guidelines*. Houston, TX, Abnormal Situation Management (ASM).

Vedoucí práce:

Ing. Leoš Beran, Ph.D.
Ústav mechatroniky a technické informatiky

Konzultant práce:

Eva Doubková
Systematic a.s.

Datum zadání práce:

10. října 2019

Předpokládaný termín odevzdání:

18. května 2020

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

Prohlášení

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

Jsem si vědom toho, že na mou bakalářskou práci 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 bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci 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.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

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.

1. června 2020

Tomáš Kubíček

Abstrakt

Tématem bakalářské práce je vytvoření grafického rozhraní pro servisního technika skladovacího robota. Skladovací robot je součástí skladovacího systému SysLogeum 3000 firmy Systematic a.s. Cílem této práce je vytvoření rozhraní, které bude servisnímu technikovi umožňovat obsluhu robota. Práce popisuje skladovací systém, pro který je rozhraní určeno, a jeho výhody a nevýhody. Dále je zde uveden vývoj grafických rozhraní v průmyslu a jejich využití v dnešní době. V práci je dále popsána technologie mapView a stručné návody, jak používat jednotlivé funkce a komponenty této technologie. V poslední části práce je popsán návrh rozhraní a jednotlivých stránek použitých ve vizualizaci, jejich účel a způsob ovládání.

Klíčová slova:

Vizualizace, grafické rozhraní, vizuální prvek, mapView, webové technologie.

Abstract

The topic of the bachelor's thesis is the creation of a graphical interface for a service technician of a storage robot. The storage robot is part of the SysLogeum 3000 storage system from Systematic a.s. The aim of this work is to create an interface that will allow the service technician to operate the robot. The work describes the storage system for which the interface is intended, and its advantages and disadvantages. The development of graphical interfaces in industry and their use today is presented. The work also describes the mapView technology and brief instructions how to use the individual functions and components of this technology. The last part of the thesis describes the design of interfaces and individual pages used in visualization, their purpose and method of control.

Key words:

Visualization, graphical interface, widget, mapView, web technology.

Poděkování

Tímto bych rád poděkoval svému vedoucímu práce Ing. Leoši Beranovi, Ph.D. za potřebnou pomoc a rady, které mi poskytl. Poděkovat bych chtěl i studentům Bc. Tomáši Myslovcovi a Bc. Jiřímu Kulichovi za jejich nápady možných řešení použité v práci. Dále bych rád poděkoval mým rodičům Ing. Martinu Kubíčkovi a Haně Kubíčkové za pomoc a podporu při psaní této práce.

Obsah

Seznam zkratek	9
1 Úvod	10
2 Výhody a nevýhody skladovacího systému SysLogeum 3000	12
3 Rozhraní mezi člověkem a strojem	15
4 Seznámení s MappView	17
4.1 Struktura HMI aplikace	18
4.1.1 Logický přehled (Logical View)	18
4.1.2 Konfigurační přehled (Configuration View)	18
4.1.3 Editor obsahu stránky (Content Editor)	19
4.1.4 Panel nástrojů (Toolbar)	20
4.1.5 Katalog (Widget Catalog)	20
4.1.6 Vlastnosti (Properties window)	20
4.2 SVG obrázky	20
5 Vytvoření vizualizace	22
5.1 Vytvoření stránky	22
5.2 Vytvoření Obsahu stránky (Content)	24
6 Vizualní prvek (Widget)	25
6.1 Textový systém (Text system)	25
6.2 Navázání prvku s proměnnou (Binding)	26
6.2.1 Proměnná v programu	26
6.2.2 Session proměnná	27
6.2.3 Výrazy (Expressions)	27
6.3 Akce a události (Events and actions)	29
7 Rozložení vizualizace (Layout)	30
7.1 Hlavička (HeaderContent)	30
7.1.1 Servisní ovládání	31
7.2 Navigace (NavigationContent)	33

8	Návrh vizualizace pro skladovacího robota	34
8.1	Přihlášení (Login)	34
8.2	Přehled (Overview)	35
8.2.1	Hardware	36
8.2.2	Motory	37
8.2.3	Deska	37
8.2.4	Plošina	38
8.3	Ovládání robota	39
8.4	Servis (Service)	40
8.5	Avatar	42
8.6	Baterie (Battery)	42
8.7	Alarmy (Alarms)	43
8.8	Tester	44
9	Závěr	48
	Seznam použité literatury	50
	Přílohy	51
A	Porovnání předchozí a nové vizualizace	51
B	Maják	59
B.1	Tlačítko na Majáku	59
B.2	Tlačítka a signalizace pro řízení pohybu v servisním režimu	61
C	Návod	63

Seznam zkratek

CSS	Jazyk pro popis způsobu zobrazení prvků na webových stránkách (Cascading Style Sheets)
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
GIF	Grafický formát určený pro rastrovou grafiku (Graphics Interchange Format)
HMI	Rozhraní mezi člověkem a strojem (Human Machine Interface)
HTML	Značkovací jazyk pro vytváření webových stránek (Hypertext Markup Language)
HTTP	Internetový protokol určený pro komunikaci s webovými servery (Hypertext Transfer Protocol)
HW	Hardware
ID	Identifikátor (Identity)
IP	Protokol používaný v počítačových sítích a internetu (Internet Protocol)
LED	Elektroluminiscenční dioda (Light Emitting Diode)
OPC UA	Otevřený standard pro komunikaci mezi strojem a webovým serverem (Open Platform Communications Unified Architecture)
PC	Osobní počítač (Personal Computer)
PLC	Programovatelný logický automat (Programmable Logic Controller)
SVG	Značkovací jazyk a formát souboru, který popisuje dvojrozměrnou vektorovou grafiku (Scalable Vector Graphics)
TUL	Technická univerzita v Liberci
XML	Značkovací jazyk (Extensible Markup Language)

1 Úvod

Tématem této práce je vytvoření grafického rozhraní pro servisního technika skladovacího systému SysLogeum 3000. Rozhraní bude technikovi umožňovat vzdálené ovládání skladovacího robota.

Skladovací systém SysLogeum 3000 přináší nové řešení pro skladování zboží. Zboží je uloženo v přepravekách. Tyto přepravky je možné naskládat na sebe do komínů. Jednotlivé komíny z přepravek jsou umístěny těsně k sobě. Konstrukce skladu pak má tvar kvádru, jehož velikost je závislá na velikosti, tvaru a počtu přepravek. K přepravekám naskládaným na sebe je možné přistupovat pouze ze shora. Z tohoto důvodu je horní část konstrukce skladu pokryta kolejnicemi. Tyto kolejnice vedou nad mezerami mezi komíny přepravek. Při pohledu shora kolejnice vytváří mříž. Této mříži se říká grid. Kola robota jsou vytvořena tak, aby zapadala do kolejnic gridu. To umožňuje robotům pohyb po celé ploše skladu a přístup k jakémukoliv komínu. Robot je vybaven pohyblivou deskou s uchopovacím mechanismem. Touto deskou může robot pohybovat nahoru a dolů, uchopit nejvýše umístěnou přepravku a přemístit ji.

Vzhledem k tomu, že robotů může být ve skladu několik a každý z nich se může pohybovat v libovolné části skladu, není možné se k jednotlivým robotům připojovat přímo pomocí kabelu. K robotům je možné se připojit pouze bezdrátově. To omezuje výběr rozhraní, které je možné použít.

Firmware robota je vytvořen ve vývojovém prostředí Automation Studio od firmy B&R. Toto prostředí má v sobě zabudovanou technologii mapView. Tato technologie umožňuje vývojáři vytvořit webovou vizualizaci. Samotná vizualizace běží na serveru, který vytváří PLC robota. K této vizualizaci je možné se připojit prostřednictvím libovolného chytrého zařízení (zařízení s přístupem na internet). Výhodou této technologie je možnost použití stejného vývojového prostředí jak pro grafické rozhraní, tak pro firmware. Pokud na projektu pracuje více vývojářů, je díky tomu jejich spolupráce jednodušší. Další výhodou je, že pro připojení k rozhraní není potřeba konkrétního zařízení od firmy B&R. Díky těmto výhodám byla jako grafické rozhraní pro tuto práci určena webová vizualizace.

Tato práce navazuje na bakalářský projekt, ve kterém byla vytvořena vizualizace pro první prototyp robota. Nová vizualizace bude sloužit k ovládání další verze skladovacího robota. Bude muset mít upravené rozhraní v závislosti na nové konstrukci robota. Je zapotřebí, aby splňovala požadavky provozovatele. Tyto požadavky se budou odvíjet od získaných praktických zkušeností s předchozí verzí vizualizace.

Výsledná práce bude nasazena do reálného skladu a testována provozovatelem systému.

Druhá kapitola této práce popisuje výhody a nevýhody skladovacího systému SysLogeum 3000. Podrobnější popis grafických rozhraní a technologie mapView je obsažen ve třetí a čtvrté kapitole. V páté a šesté kapitole je popsán postup při vytváření vizualizace, možnosti mapView a výhody a nevýhody této nové technologie. Sedmá kapitola popisuje oblasti nové vizualizace, prvky, které jsou k dispozici na všech stránkách, a výhody nové verze oproti předchozí. Podrobnější popis jednotlivých stránek a rozdílů oproti předchozí verzi vizualizace je popsán v osmé kapitole.

2 Výhody a nevýhody skladovacího systému SysLogeum 3000

V této kapitole je možné zjistit jakými výhodami a nevýhodami disponuje skladovací systém SysLogeum 3000. Jsou zde zmíněny i další firmy, které se zabývají tímto druhem skladovacího systému.



Obrázek 2.1: Skladovací systém Archeion [1]

Nový skladovací systém SysLogeum 3000 přináší mnoho výhod do celé řady oborů.

- Z hlediska prostorové využitelnosti skladu je tento systém nejefektivnější. Převrsky je možné uskladňovat těsně k sobě bez potřeby vytvoření mezery mezi nimi pro průchod lidí nebo průjezd manipulační techniky. Díky tomu je možné využít efektivněji vnitřní prostor skladu. [2]
- Z hlediska technického je omezen vliv lidského faktoru na přepravu přepravky. Ve skladu využívající tento systém by nemělo dojít k upuštění přepravky, uložení přepravky na špatnou pozici nebo nehodám způsobených špatnou manipulací s přepravkou.
- Přepravky včetně skladovaných produktů mohou vážit až 25 kg. Při potřebě vyskladnění několika přepravek naráz se tak může jednat o dost namáhavou práci. Robot je konstruován tak, aby neměl problém takto těžké přepravky zvedat a pokládat cyklicky celý den a noc. Z pohledu bezpečnosti ochrany zdraví při práci se tak jedná o pokrokový systém.

- Sklad, pro který je tento systém vyvíjen, je určen ke skladování dokumentů. Pokud by u běžného skladu došlo k nehodě, která by způsobila požár, jsou veškeré dokumenty uloženy ve skladu ohroženy. Podmínkou tohoto systému je nutnost dokumenty ukládat do přepravek, které jsou následně naskladněny na sebe. Tyto přepravky tak oddělují jejich obsah a v případě požáru jsou dokumenty uvnitř nich chráněny. Po stránce ochrany dokumentů se tak jedná i o bezpečný systém. Teoreticky je možné umístit sklad do hermeticky izolovaného prostředí. Kdyby se v tomto prostředí upravovala koncentrace kyslíku ve vzduchu, nemohlo by dojít k požáru.
- Další výhody přináší tento systém do odvětví ekonomiky provozu. Jeden z největších provozních nákladů skladu je osvětlení a vytápění. Roboti ke své činnosti teplo ani světlo nepotřebují. Pokud se sklad nenachází v nepříznivých teplotních podmínkách, na které roboti nejsou stavěny, je možné sklad provozovat bez osvětlení a vytápění. Tím se sníží náklady na provoz skladu. [2]

Na druhou stranu existují i nevýhody tohoto systému.

- Hlavní nevýhodou jsou počáteční náklady na vývoj, sestavení skladu a konstruování robota. Aby sklad fungoval bezchybně, musí být pečlivě zkonstruován. Každá nedokonalost při sestavování skladu může vést časem k problémům. Tyto problémy mohou vést k prodloužení potřebné doby pro vyskladnění (naskladnění) nebo k chybám robota, které musí řešit servisní technik nebo administrátor skladu.
- Další nevýhodou je složitější logika řízení robota nebo i robotů. Tyto sklady nejsou určeny jen pro jednoho robota. Při pohybu několika robotů naráz nesmí dojít k jejich kolizi. K tomu je potřeba vytvoření nadřazeného plánovacího systému (plánovače), který bude plánovat celý sklad, řídit roboty a dopravníky. [2]
- V případě potřeby vyskladnit nejnižše položenou přepravku je nutné přeskládat všechny přepravky nad ní, což značně způsobuje zpomalení vyskladnění dané přepravky.

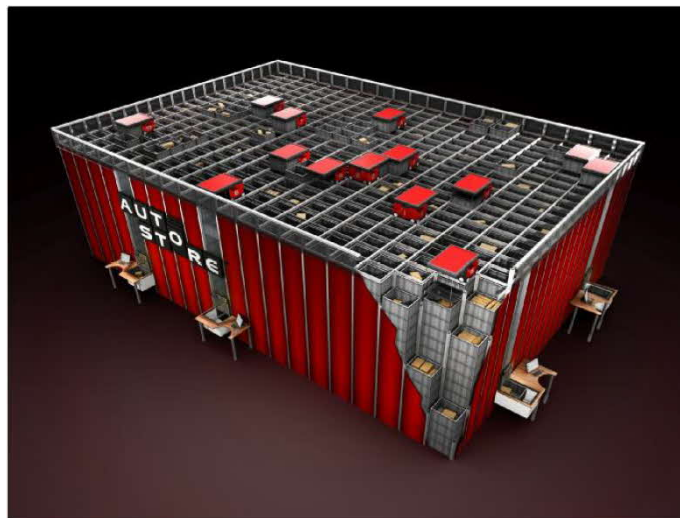
Firma Ocado, která též vyvinula tento typ skladu, částečně vyřešila tuto situaci tím, že přepravky, které se často vyskladňují, jsou umístěny na vyšších pozicích. Přepravky, které se vyskladňují méně často, jsou umístěny níže. Tímto způsobem se jim podařilo snížit potřebný čas pro vyskladnění. Navíc pokud je potřeba vyskladnit nejnižše položenou přepravku, začnou volní roboti spolupracovat. Ve skladu firmy Ocado je možné na sebe naskládat až 17 přepravek. Při spolupráci několika robotů je vyskladnění takové přepravky mnohem rychlejší. [3]

- Poslední nevýhoda se týká spíše použitých dopravníků pro přesun přepravky od robota k uživateli než skladovacího systému. Sklad firmy Systematic a.s. má

v tuhle chvíli k dispozici tři dopravníky. Po přijmutí požadavku na vyskladnění robot přiveze přepravku na dopravník a dopravník ji předá na pozici pro uživatele. Z této pozice ji uživatel musí buď vyjmout, nebo opět naskladnit. V praxi to znamená, že v jednu chvíli je možné vyskladnit naráz pouze tři přepravky. Pro vyskladnění další je potřeba uvolnit jeden z dopravníků. Pro vyskladnění a naskladnění většího počtu přepravek je zapotřebí přítomnost operátora. Do budoucna se plánuje vývoj nového typu dopravníku, který by umožnil naskladnit nebo vyskladnit víc přepravek. Celý dopravník by měl tři úložné prostory (zásobníky) pro přepravky. Jeden by obsahoval přepravky určené k naskladnění, druhý k vyskladnění a třetí přepravky, které nemohly být naskladněny z důvodu chyby (špatné sériové číslo přepravky nebo nadměrná hmotnost).

Všechny výše zmíněné nevýhody je možné během vývoje nebo i za provozu skladu časem eliminovat. Po vložení potřebného času a peněz se tak jedná o velice efektivní sklad téměř po všech stránkách.

Tímto skladovacím systémem se kromě TUL a Ocado zabývá ještě firma AutoStore. Sklad využívající tento systém nazval americký technický zpravodajský portál The Verge (2018): „AUTOMATED WAREHOUSE OF THE FUTURE” (autonomním skladem budoucnosti). [3]



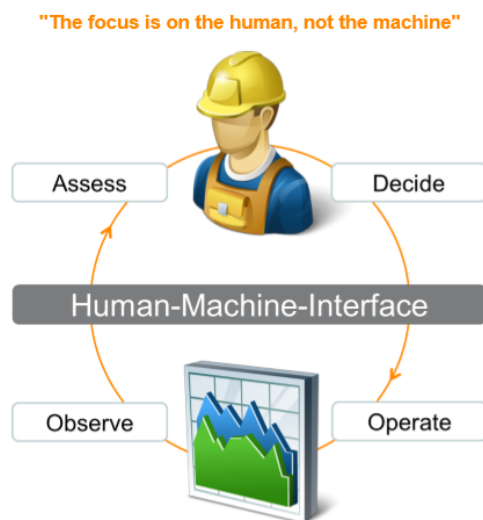
Obrázek 2.2: Skladovací systém Autostore. [3]

3 Rozhraní mezi člověkem a strojem

Vývoj grafických rozhraní, jejich podoba v dnešní době, co je technologie mapView, to vše je možné zjistit v této kapitole.

HMI (Human Machine Interface) nebo-li rozhraní mezi člověkem a strojem, je určeno ke kontrolování a ovládání daného stroje. Dříve musel operátor jít a osobně zkontrolovat a zaznamenat stav, ve kterém se stroj nachází. Díky real-time komunikaci mezi PLC a HMI není již potřeba, aby operátor musel osobně kontrolovat jeden stroj. Může jich dokonce kontrolovat hned několik z jednoho místa. Tím se sníží čas potřebný pro diagnostiku problému a eliminují se tak chyby způsobeny nedostatkem informací nebo chybou operátora. [4]

Existuje spousta druhů HMI. Dřívější HMI bylo pouze ovládacím panelem s tla-



Obrázek 3.1: Komunikace člověk-stroj [5]

čítky. Některé lepší stroje mohly mít i displej s jednoduchou vizualizací. Jednoduché vizualizace mohly sloužit k zobrazení stavu vstupů a výstupů. Složitější umožňovaly stroj vypínat nebo měnit rychlost produkce. Postupem času se HMI vylepšovalo a s technologií dotykových displejů a lepšího rozlišení bylo možné vytvořit rozhraní více intuitivní a uživatelsky přívětivější. [4]

V dnešní době je možné vytvořit vizualizaci pro chytrá zařízení prostřednictvím webové aplikace. Uživatel tak může ovládat stroj pomocí chytrého mobilu nebo tabletu. Tato aplikace umožňuje operátorům okamžitý přístup k informacím o stroji

z kteréhokoliv místa s připojením k lokální síti. Pokud má uživatel přístup k lokální síti i z jiné sítě (např. domácí), může stroj ovládat bez nutnosti osobní přítomnosti ve skladu (např. mimo pracovní dobu). Díky tomu je možné kontrolovat chod skladu i ze vzdálených míst. [4]

Jednou z technologií, umožňující vytvořit tento druh vizualizace, je technologie mappView. Hlavní výhodou této technologie je možnost vytvoření webové vizualizace bez znalosti webových programovacích jazyků. Při práci s kteroukoliv jinou technologií pro vytváření webových aplikací je znalost těchto jazyků nezbytná. mappView je díky tomu první technologií na světě, se kterou je možné vytvořit webovou vizualizaci bez potřeby znát tyto jazyky. Navíc je plně integrovaná přímo ve vývojovém prostředí AS. Není tedy potřeba při vytváření firmwaru a vizualizace stroje využívat dalších prostředí. Další výhodou je, že pro otevření vytvořené vizualizace je možné použít jakékoliv chytré zařízení s webovým prohlížečem a přístupem k internetu. Uživatel není omezen pouze na zařízení od firmy B&R, ale může si zvolit kterékoliv jiné zařízení od kteréhokoliv výrobce (Apple, Samsung aj.). [6]

Jak jsem zmínil výše: „dřívější HMI bylo pouze ovládacím panelem s tlačítky“, takto bych popsal vizualizaci vytvořenou v rámci projektu. [7] Většinu komponent tvořila tlačítka a jednoduché indikátory stavů. Při tvorbě nové vizualizaci šlo především o zachování funkcionalit použitých v projektu, které se osvědčily a již jsou na ně pracovníci ve skladu zvyklí. Nová vizualizace by měla splňovat požadavky pro 21. století. Hlavním požadavkem je intuitivní ovládání. Oproti projektu jsou v nové vizualizaci použity SVG obrázky (obrázky využívající vektorovou grafiku), které jsou určeny pro webové vizualizace. Tyto obrázky by měly operátorovi zjednodušit diagnostiku a ovládání robota.

Při popisu jednotlivých stránek vizualizace jsou porovnávány rozdíly mezi předchozí a novou vizualizací. V přílohách je pak možné tyto stránky vidět vedle sebe.

4 Seznámení s MappView

Podrobnější informace ohledně technologie mappView, její výhody a nevýhody jsou popsány v této kapitole. Součástí této kapitoly je i stručný popis webových technologií a SVG obrázků.

Technologie mappView je založena na značkovacím jazyce HTML5, CSS3 a JavaScriptu. AS umožňuje vytvoření HMI a logiky stroje a oddělení jejich obsahu. Díky tomu existuje možnost opakovaného použití stejného HMI pro různé stroje. Požadovaná znalost webové technologie není potřebná, protože je zapouzdřena ve vizuálních prvcích (widget), které lze jednoduše přidávat a konfigurovat podle potřeby. Prvkem vizualizace může být např. tlačítko nebo obrázek. Práce s prvky je jednoduchá. Vývojář se tak při vytváření vizualizace může soustředit především na její vzhled a funkcionalitu. [6]



Obrázek 4.1: mappView HMI aplikace. [6]

To, co je hlavní výhodou mappView, je současně i nevýhodou. Webové technologie jsou opravdu zapouzdřené ve vizuálních prvcích. Bohužel se k těmto technologiím ve většině případů nelze dostat a jednotlivé prvky tedy není možné upravovat dle možností použitých technologií HTML5 a CSS3. MappView umožňuje vývojáři provádět některé úpravy grafického stylu prvku. Grafické styly jsou pospány pomocí CSS3. Úpravou grafického stylu může být např. změna barvy prvku, okrajů nebo barvy textu v prvku. Vývojář má i možnost vytvořit zcela nový grafický styl prvku.

Tady se vývojář sice dostává na úroveň webové technologie CSS3, ale má jen omezené možnosti při psaní stylu.

Některé vlastnosti je možné navázat na proměnné a tím upravovat např. vzhled prvku za běhu programu. I v tomto případě je vývojář omezen tím, pro které vlastnosti a prvky toto navázání mapView umožnilo. Při práci s objektově orientovanými programovacími jazyky (např. C#) je možné objekty libovolně v programu upravovat, měnit jejich vlastnosti nebo jim i vlastnosti přidávat. Jako hlavní rozdíl bych uvedl možnost v C# měnit polohu objektu (prvku) přímo z programu. V současnosti mapView tuto schopnost nemá.

Pokud chce vývojář udělat vizualizaci více interaktivní, může použít funkce *Akce a události*¹ (Events and actions). Tyto funkce umožňují reagovat na události (např. kliknutí na obrázek) vyvoláním dané akce. Zde se už mapView touto funkcí trochu přibližuje objektově orientovanému programování, i když k tomu využívá zcela jiných technologií. Pro přesnější popis viz kapitolu 5.

Správa dat v mapView je zcela založena na OPC UA serveru. To umožňuje snadno integrovat zdroje dat ze třetích stran do HMI. Nejedná se tedy pouze o komunikaci mezi vizualizací a PLC. K OPC UA serveru se může připojit a upravovat data kterékoliv jiné zařízení, mající k němu přístup. V případě skladu je třetí stranou plánovač, který přes OPC UA server předává instrukce robotům a dopravníkům. Tato data mohou být prohlížena lokálně na panelu PC nebo vzdáleně ze standardního webového prohlížeče pomocí vizuálních prvků. Je tak možné, ale ne příliš efektivní, ke kontrole využít pouze OPC UA server a přímo sledovat stavy jednotlivých proměnných. [6]

4.1 Struktura HMI aplikace

Vlastní vizualizace se tvoří v prostředí AS. Na Obr. 4.2 je zobrazeno okno, pomocí kterého se vizualizace vytváří. Veškeré prvky aplikace jsou v tomto okně rozděleny do několika sekcí. Tyto sekce obsahují části celého projektu. Níže je popsán pouze obsah, který je potřebný pro vizualizaci.

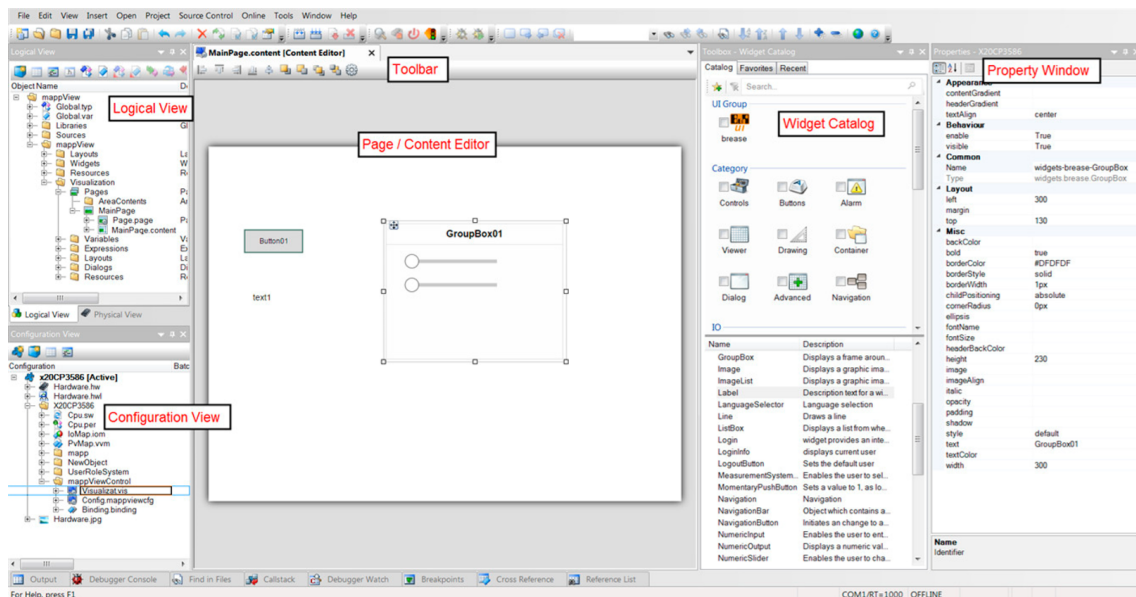
4.1.1 Logický přehled (Logical View)

Logický přehled obsahuje vizuální části aplikace (jednotlivé stránky, texty, mediální soubory atd.). [6]

4.1.2 Konfigurační přehled (Configuration View)

V *Konfiguračním přehledu* se konfiguruje jedna nebo i více HMI aplikací. Vývojář má možnost konfigurovat např. maximální počet připojených klientů, obnovovací frekvenci vizualizace, ID a další vlastnosti vizualizace. Nastavuje se zde propojení

¹Názvy funkcí a souborů, které je možné použít v mapView jsou přeloženy do češtiny a označeny tímto způsobem: první písmeno je velké, text je zkosený a u prvního výskytu je v závorce originální název.



Obrázek 4.2: Struktura HMI aplikace v AS. [6]

prvků s proměnnými pomocí *Listu navázání dat* (Binding list) nebo *Akcí a událostí*. A povolují se zde proměnné, které mají být k dispozici na OPC UA serveru. [6]

4.1.3 Editor obsahu stránky (Content Editor)

Jak název napovídá, *Editor obsahu stránky* je vizuální zobrazení obsahu stránky. Veškeré prvky jsou umístěny v *Obsahu stránky* (Content). *Obsah stránky* je dílčí částí stránky. Chce-li vývojář zobrazit nějaký prvek ve vizualizaci, musí vytvořit *Stránku* (Page), definovat její *Rozložení* (Layout) a navázat *Obsah stránky* s oblastí na stránce pomocí jeho ID viz níže. *Obsah stránky* je možné otevřít i pomocí textového editoru, který obsahuje XML kód. XML kód je značkovací jazyk, který popisuje obsah stránky. Jednotlivé prvky na stránce jsou elementy tohoto jazyka. XML kód je skutečná podoba stránky. To, že má vývojář k dispozici její vizuální zobrazení, je pro něj obrovská výhoda, protože mu to umožňuje vyhnout se psaní XML kódu. Veškeré akce (přidání, přesunutí nebo upravení prvku) v *Editoru obsahu stránky* MappView převádí do XML kódu automaticky. Nemusí se tedy *Obsah stránky* programovat pomocí kódu, jako to je u některých aplikací. Za normálních okolností většinou není potřeba otevírat textový editor *Obsahu stránky*. Je vhodné ho otevřít v případech, kdy při vytváření stránky provede vývojář nějakou chybu. Např. přesune prvek mimo *Obsah stránky* tak, že ho není možné přesunout zpátky nebo pokud je nějakým způsobem poškozena syntaxe kódu, nemusí grafický editor vždy zkonstruovat vzhled *Obsahu stránky*. [6]

4.1.4 Panel nástrojů (Toolbar)

Panel nástrojů slouží k dizajnování prvků (např. přesunutí prvku do popředí nebo do pozadí). Je zde i možnost úpravy Mřížky (Grid) použité v *Editoru obsahu stránky*. K této Mřížce jsou automaticky přichycovány prvky umístěné v *Editoru obsahu stránky*. [6]

4.1.5 Katalog (Widget Catalog)

Když je některý *Obsah stránky* otevřený (graficky nebo textově), je možné do něj přidat prvek z *Katalogu* a konfigurovat ho. Přidání prvku je umožněno metodou „táhnout a pustit“ (drag and drop). Touto metodou se v kódu *Obsahu stránky* vytvoří řádek nesoucí informace o vlastnostech prvku (velikost, umístění aj.). Je taky možné tento řádek připsat ručně. [6]

4.1.6 Vlastnosti (Properties window)

Elementy HMI aplikace jsou konfigurovány zde. V závislosti na prvku jsou k dispozici jiná editovací okna. I tyto *Vlastnosti* jsou v pozadí aplikace jen jako kód. Změnou kterékoliv z nich se přepíše příslušný řádek s prvkem. Prvek je popsán pomocí několika kódů. Pokud se jedná o úpravu vzhledu, je přepsán kód CSS3. Pokud se jedná o změnu proměnné, na kterou byl prvek navázán, je přepsán kód XML. [6]

4.2 SVG obrázky

„SVG (z anglického Scalable Vector Graphics škálovatelná vektorová grafika) je značkovací jazyk a formát souboru, který popisuje dvojrozměrnou vektorovou grafiku pomocí XML. Formát SVG je základním otevřeným formátem pro vektorovou grafiku na webových stránkách. HTML5 umožňuje vložit kód SVG obrázku přímo do kódu HTML webové stránky.” [8] Obrázek je pak možné zobrazit pomocí prohlížeče.

I v případě HTML5 se jedná o značkovací jazyk. Tento jazyk se skládá ze značek, které popisují obsah webu (obdobně jako XML v technologii mapView). Tento jazyk se nezabývá vzhledem webové stránky. K tomuto účelu slouží jazyk CSS3. HTML5 umožňuje vložit do kódu odkaz na soubor CSS3, který popisuje vzhled stránky. [9]

Stejně jako u HTML5 je možné aplikovat i na SVG kaskádní styly CSS3 nebo JavaScript. Pomocí CSS3 se dá SVG graficky upravovat nebo z obrázku udělat animaci. JavaScript pak umožňuje udělat obrázek interaktivní (např. vytvoření akce po kliknutí na obrázek). [8]

Rastrová grafika zobrazuje obrázek pomocí jednotlivých pixelů (nejmenší možný zobrazený bod na displeji). Vektorová grafika k zobrazení používá objekty (např. čára nebo kruh) a jejich vlastnosti (velikost, barva, poloha atd.). Rozdíl mezi obrázky využívající rastrovou grafiku a vektorovou grafiku je možné vidět po přiblížení obrázku v prohlížeči. Obrázek využívající vektorovou grafiku se bude jevit pořád

stejně. Na obrázku využívající rastrovou grafiku budou viditelné jednotlivé pixely (především u nerovných hran). [8]

Nejčastější potřeba přiblížení stránky je u mobilních aplikací. Mobilní telefony nemají ani zdaleka tak velký displej jako mají počítače. Proto uživatel často některé části stránky přibližuje a oddaluje. Z tohoto důvodu se vektorová grafika využívá u webových aplikací, aby bylo možné aplikaci zobrazovat nejen pomocí prohlížeče v počítači, ale i pomocí prohlížeče v chytrém zařízení.

Náročnost vektorové grafiky se odvíjí od množství objektů. Čím detailnější je obrázek, tím více objektů je potřebných k jeho popisu. Při každé změně přiblížení se znova přečte popis obrázku (SVG formát) a obrázek se překreslí. U složitějších (detailnějších) obrázků by toto překreslení mohlo trvat dlouho a zbytečně zvyšovalo náročnost aplikace. Proto jsou obrázky využívající vektorovou grafiku často jednoduché.

MappView umožňuje navázat SVG obrázek na dva prvky. Jedním je prvek *Papír* (Paper), který k obrázku přistupuje jako ke značkovacímu jazyku a je s jeho pomocí možné obrázek i animovat (viz kapitolu 8). Druhým je prvek *Obrázek* (Image), který SVG formát pouze graficky zobrazuje.

Díky vlastnostem SVG formátu, které jsou vhodné pro webovou vizualizaci, byl tento formát použit pro všechny obrázky obsažené v této vizualizaci. Téměř všechny obrázky byly pro tuto vizualizaci vytvořeny v prostředí InkScape. Toto prostředí je určeno k vytváření SVG obrázků a je preferováno pro použití v nápovědě AS.

5 Vytvoření vizualizace

Tato kapitola obsahuje stručný postup pro vytvoření vizualizace pomocí technologie `mappView`. Jsou zde popsány jednotlivé kroky pro vytvoření stránky a rady, které usnadňují vytváření vizualizace.

Chce-li vývojář vytvořit v projektu vizualizaci, musí nejprve do *Logického přehledu* přidat soubor *MappView*. Tento soubor obsahuje veškeré prvky vizualizace (stránky, dialogy, styly, texty, obrázky, jazykové mutace atd.) kromě konfiguračních souborů (navázání prvků s daty nebo akcí a událostí). Tyto soubory se nacházejí v *Konfiguračním přehledu*. Po kliknutí na soubor *MappView* je možné přidat z *Katalogu* soubor *Vizualizace* (Visualization). V něm se nachází složka *Stránky* (Pages). Do této složky je možné přidat stránky a jejich obsah. Důležité je si uvědomit, že po přidání jakékoliv části vizualizace se automaticky vygeneruje její ID. Název souboru a ID souboru nejsou shodné, což může být ze začátku dost matoucí a může to způsobovat problémy při kompilaci programu. Z tohoto důvodu je doporučeno (a podle mne i rozumné), aby byl název a ID stejný. Vytváření vizualizace je pak přehlednější, lehčí a eliminuje to možné komplikace při zadávání ID. [6]

Téměř všechny části vizualizace, které definují její vzhled (*Rozložení*, *Obsah stránky*, *Dialogy*) využívají pro svůj popis značkovací jazyk XML. Ačkoliv je tento jazyk původně určený pro výměnu dat a databáze, v `mappView` se používá pro popis částí vizualizace. Grafické editory umožňují zobrazit vzhled stránky na základě tohoto jazyka a graficky jej upravovat. Opět není znalost tohoto jazyka nezbytná. Pouze při vytváření *Akcí a událostí* a návrhu *Rozložení* jej vývojář musí ovládat. Firma B&R pracuje na grafickém editoru pro tyto funkce. Ten v této chvíli zatím není k dispozici. [6]

5.1 Vytvoření stránky

Před přidáním stránky z *Katalogu* je potřeba nejprve vytvořit *Rozložení*. Toto *Rozložení* definuje velikost stránky a oblasti, do kterých se přidá požadovaný *Obsah stránky*. Pro přiřazení *Rozložení* ke *Stránce* musí znát vývojář ID *Rozložení*. To může zjistit otevřením příslušného *Rozložení*. Na druhém řádku kódu najde atribut ID. Pokud se název souboru shoduje s ID, stačí jen zkopírovat jeho název a není potřeba otevírat tento soubor a hledat ID v něm. Vzhledem k tomu, že *Rozložení* není zobrazením stránky, ale jen jejím uspořádáním, je možné jej otevřít pouze v textové podobě viz Obr. 5.1. Umístění každé oblasti se odvíjí od levého horního okraje stránky. [6]



Obrázek 5.1: Rozložení (Layout)

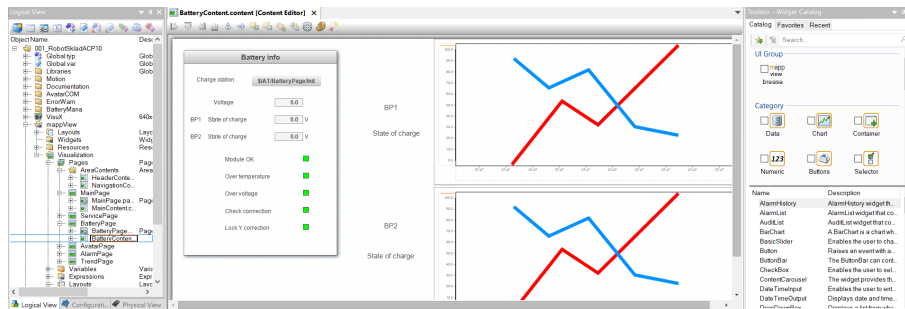
Po vytvoření *Rozložení* je možné do souboru *Stránky* (Pages) přidat *Stránku* (Page). *Stránka* obsahuje dva soubory, jeden je s příponou *.page*, druhý má příponu *.content*. Soubor s příponou *.page* obsahuje informace o *Rozložení* a přiřazuje *Obsah stránky* do oblastí definovaných *Rozložení*m viz Obr. 5.2. Zároveň definuje barvu pozadí každé oblasti nebo je možné umístit do pozadí stránky obrázky. Soubor s příponou *.content* je *Obsah stránky*. Tento soubor by měl být hlavní oblastí stránky. Pouze *Stránka*, v jejíž složce se tento soubor nachází, by měla být navázána na tento *Obsah*. Je totiž možné k oblasti na stránce přiřadit jakýkoliv *Obsah stránky* i přesto, že se nenachází ve stejném souboru *Stránka*. To opět může způsobovat komplikace při vytváření vizualizace. Pro použití stejného *Obsahu* na více stránkách, existuje složka *Oblast obsahů* (AreaContents). Ta se nachází ve složce *Stránky* nad složkami jednotlivých stránek a je určena k této činnosti. Chce-li vývojář pak najít určitý *Obsah stránky*, nemusí ho hledat podle jeho ID nebo názvu, protože ví, kde přesně se nachází. [6]



Obrázek 5.2: Struktura stránky. [6]

5.2 Vytvoření Obsahu stránky (Content)

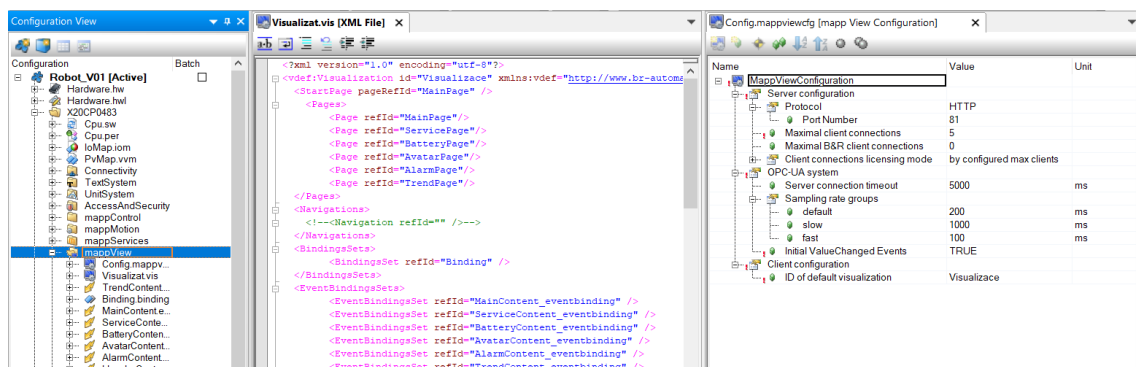
Po přidání souboru *Obsah stránky* (Page content) z *Katalogu* je možné definovat jeho velikost a prvky, které na něm budou umístěny. Aby se vytvořený *Obsah* zobrazil na příslušné stránce tak, jak je vidět v *Editoru obsahu stránky*, musí mít stejnou velikost jako oblast rozložení, pro kterou je určen. Jinak by mohlo dojít k oříznutí *Obsahu stránky*. Po splnění této podmínky se vývojář může výhradně soustředit na přidávání a konfigurování prvků vizualizace. [6]



Obrázek 5.3: Obsah stránky (Content)

To poslední, co je potřeba k zobrazení stránky udělat, je do *Konfiguračního přehledu* přidat soubor *MapView*, do kterého se musí přidat soubory *Visualizat.vis* a *Config.mapviewcfg*. Soubor *Visualizat* obsahuje odkazy (ID) na všechny soubory, které jsou potřeba ve vizualizaci (stránky, navigace, navázání, dialogy atd.). Například, aby bylo propojení prvku s proměnnou pomocí *Listu navázání* funkční, musí se do souboru *Visualizat* přidat odkaz na tento soubor. Tento soubor také obsahuje samotné ID vizualizace, což umožňuje otevřít vizualizaci v prohlížeči. V souboru *Config* se konfiguruje vizualizace. [6]

Po nastavení všech souborů je možné se k vizualizaci připojit pomocí webového prohlížeče. Adresa vizualizace se skládá z IP adresy zařízení, ve kterém je vizualizace nahrána (simulátor, PLC), čísla portu vizualizace (standardně port 81) a názvu vizualizace (ID vizualizace v souboru *Visualizat*). Výsledná adresa pak může vypadat takto: `127.0.0.1:81/Vizualizace`.



Obrázek 5.4: Ukázka souboru Visualizat a Config

6 Vizuální prvek (Widget)

Obsahem této kapitoly je popis prvků vizualizace, jejich možností a způsob použití v `MapView`.

Vizuální prvek je jakýkoliv prvek umístěný v *Obsahu stránky*. Existuje mnoho prvků (textové prvky, prvky pro zobrazení hodnot proměných, obrázky atd.). Každých z těchto prvků má jiné vlastnosti a jiné možnosti navázání nebo jiné *Akce a události*. Všechny prvky se nachází v *Katalogu*. Jejich popis je možné dohledat v nápovědě AS. Prvky jsou rozděleny do několika kategorií, pomocí nichž je možné tyto prvky filtrovat. Chce-li vývojář např. zjistit, které prvky může použít pro zobrazení textu, stačí zaškrtnout kategorii *Text* a v *Katalogu* zůstanou jen prvky této kategorie např. *Popisek* (Label), *Zobrazení textu* (TextOutput), *Zadání textu* (TextInput) atd. Každý prvek, jenž umí zobrazit nějaký text, má vlastnost *Text* nebo *Label*, do které je možné zapsat požadovaný text k zobrazení.

6.1 Textový systém (Text system)

`MapView` poskytuje možnost vytvoření *Textového systému*. Pomocí tohoto systému je možné měnit jazyk textů celé vizualizace. Nejprve je potřeba v souboru *Jazyk projektu* (Project language) zvolit požadované jazyky. Potom se přidá do složky *Texty* (Texts) soubor *Lokalizovatelné texty* (Localizable Texts). Do tohoto souboru se zadávají texty, které se budou měnit při změně jazyka viz Obr. 6.1. K tomu, aby bylo možné propojit text prvku s tímto mutujícím textem, musí *Jmenný prostor* (Namespace) tohoto souboru začínat na IAT a každý text musí mít svoje originální *TextID*. Stále ale ještě není možné tyto texty přiřadit prvku. Ještě je potřeba do souboru *Textconfig*, který je umístěný ve složce *TextSystem* v *Konfiguračním přehledu*, přidat soubor s těmito texty. Až po splnění těchto bodů může vývojář tyto texty použít. Kliknutím na textové pole prvku ve *Vlastnostech* se v pravé části zobrazí tlačítko se třemi tečkami. Toto tlačítko otevře okno, ve kterém se nachází texty k propojení. Při zvolení nějakého textu a potvrzení tlačítkem OK se do textového pole napíše odkaz na tento text. Odkaz je tvořen *Jmenným prostorem* a *TextID* daného textu.

Text ID	Čeština (cs)	English (en)
1 Login	Přihlásit se	Login
2 Password	Heslo	Password
3 Username	Uživatel	Username
4 Logout	Odhlásit se	Logout
5 ChangePass...	Změna hesla	Change password
6 OldPassword	Staré heslo	Old password
7 NewPassword	Nové heslo	New password
8 Close	Zavřít	Close
9 User	Uživatel:	User:
10 <i>New_Text_ID</i>		

Obrázek 6.1: Lokalizovatelné texty

6.2 Navázání prvku s proměnnou (Binding)

Hlavní účel vizualizace je zobrazovat a nastavovat hodnoty a stavy proměnných. K tomu, aby daný prvek plnil tuto funkci, je potřeba provést *Navázání prvku s proměnnou*. Všechna tato propojení prvků a proměnných jsou zapisovány XML kódem do *Listu navázání* (Binding list). Tento soubor je potřeba nejprve přidat do složky *MapView* v *Konfiguračním přehledu*. Má-li nějaký prvek vlastnost *Hodnota* (Value), je možné otevřít pomocí tlačítka se třemi tečkami (obdobně jako u textů) okno obsahující všechny proměnné, které je možné s daným prvkem propojit.

6.2.1 Proměnná v programu

Jednou z proměnných, které je možné s prvkem navázat, je proměnná použitá v programu stroje. Propojení mezi prvkem a touto proměnnou není přímé. Mezi proměnnou a prvkem je prostředník, který poskytuje přístup k požadovaným datům. Tuto důležitou roli prostředníka zastává OPC UA server. Kromě přístupu k datům dané proměnné nebo prvku, je možné zde uvést, v jakých jednotkách jsou data, určit práva uživatelů k těmto datům aj. Aby byla proměnná přístupná a OPC UA server poskytoval informace o datech, musí se daná proměnná povolit. Je potřeba v *Konfiguračním přehledu* do složky *Propojení* (Connectivity) přidat soubor *OpcUaMap* z *Katologu*. Otevřením tohoto souboru se zobrazí všechny proměnné v programu. Po stisknutí pravého tlačítka myši u požadované proměnné se otevře okno, v němž se po kliknutí na *Povolit značku* (Enable Tag) zpřístupní tato proměnná.

6.2.2 Session proměnná

Takto se postupuje, pokud chce vývojář navázat proměnnou z programu s prvkem ve vizualizaci. MappView umožňuje navázat i jiné proměnné, které nemusí existovat v PLC, ale existují na serveru. Jedná se o tzv. session proměnné.

Tento typ proměnných vznikl, protože HTTP protokol neuchovává informace o stavu komunikace. [10]

HTTP zprostředkovává komunikaci mezi uživatelem (internetovým prohlížečem) a serverem. Na každý dotaz uživatele odpovídá nezávisle na předchozím, proto se mu říká bezstavový protokol. [10]

Session proměnné umožňují uchovávat informace o přihlášeném uživateli po dobu jeho připojení ke stránce. Tyto informace nejsou mezi jednotlivými stránkami sdíleny. K tomu slouží rozšíření protokolu HTTP cookie. Informace uložené v cookie (např. ID uživatele, poslední otevřená stránka apod.) jsou pak při načtení stránky prohlížečem odeslány na server. Těchto cookie souborů využívají např. internetové obchody pro uchování položek v košíku uživatele. [11]

Kosek (2000) napsal, že „session proměnné jsou jedním z nejpohodlnějších způsobů, jak obejít bezstavovost protokolu HTTP”. [12]

Použití těchto proměnných je nezbytné i pro bezpečnost vizualizace. Pokud jsou všechny proměnné ve vizualizaci navázány na proměnné v programu, každý uživatel může sledovat údaje zadávané jiným uživatelem. V případě přihlášení více uživatelů najednou je možné, aby jeden uživatel využil přihlašovací údaje druhého a přihlásil se místo něj. To je v řadě situací nepřístupné. Vzhledem k tomu, že tyto proměnné nejsou uloženy v paměti PLC, není možné k jejich propojení využít OPC UA server. Jejich použití je možné jenom prostřednictvím *Listu navázání* nebo použitím *Akcí a událostí*.

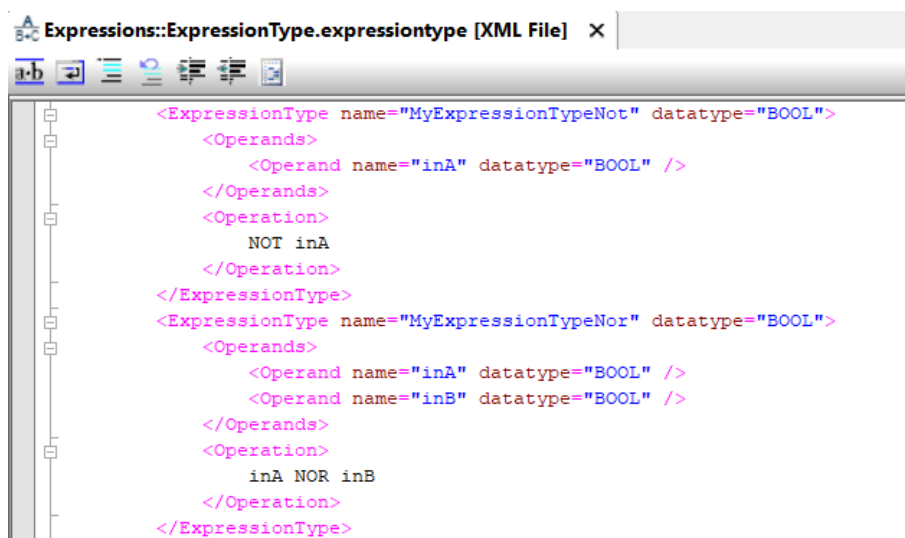
Proměnné, obsahující informace o přihlášeném uživateli, jsou generovány automaticky. Tyto proměnné jsou v mappView označeny jako *Systémové proměnné* (System variables) a je možné k nim přistupovat přes strukturu *Informace klienta* (clientInfo). Kromě těchto proměnných může vývojář vytvořit i vlastní session proměnné. Jedna taková proměnná byla v této vizualizaci použita pro zaznamenání zobrazených vlastností na stránce Přehled. Každý uživatel si tak může zobrazovat jiné vlastnosti modulů (viz kapitolu 8).

6.2.3 Výrazy (Expressions)

Další proměnnou, kterou je možné navázat na prvek, je *Výraz* (Expression). Výraz je funkce, která se vykonává na serveru vizualizace. Je vytvořen typem a instancí. Typ obsahuje informace o typu výstupu (např. binární), typ a počet vstupů (operandů) a jaké operace má Výraz se vstupy vykonat (operátory) viz Obr. 6.2. Instance pak vygeneruje Výraz na základě použitého typu viz Obr. 6.3. Vygenerovaný Výraz je přiřazen k *Obsahu stránky*. Pro jeho použití na více stránkách, na kterých je jiný *Obsah*, se musí použít instance pro každou stránku zvlášť. Je tedy možné mít jeden

typ a několik instancí Výrazu. Instance i typ jsou psány v XML kódu.

Pro použití Výrazu je potřeba přidat soubory *Typ výrazu* (ExpressionType) a *Výraz* do složky *Výrazy* (Expressions) v *Logickém přehledu* a do *Listu navázání* ručně připsat proměnné navázané na vstupy Výrazu. Soubor *Výraz* navíc obsahuje ID, které je nutné uvést v souboru *Visualizat*. Po splnění těchto podmínek je možné výsledek Výrazu navázat na prvek prostřednictvím editovacího okna přístupného přes tři tečky.



```
<ExpressionType name="MyExpressionTypeNot" datatype="BOOL">
  <Operands>
    <Operand name="inA" datatype="BOOL" />
  </Operands>
  <Operation>
    NOT inA
  </Operation>
</ExpressionType>
<ExpressionType name="MyExpressionTypeNor" datatype="BOOL">
  <Operands>
    <Operand name="inA" datatype="BOOL" />
    <Operand name="inB" datatype="BOOL" />
  </Operands>
  <Operation>
    inA NOR inB
  </Operation>
</ExpressionType>
```

Obrázek 6.2: Typ Výrazu



```
<?xml version="1.0" encoding="utf-8"?>
<ExpressionsSet id="expression_1" xmlns="http://www.br-automation.com/iat2015/expression/engineering/v3" xmlns:xsi="http://www.w
<Expressions>

  <Expression id="myNOTANDExprNLS" xsi:type="content" contentRefId="ServiceControlContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTANDExprPLS" xsi:type="content" contentRefId="ServiceControlContent" type="MyExpressionTypeNot" />

  <Expression id="myNOTExprLKNLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTExprLKPLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTExprReNLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTExprRePLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTExprFrNLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
  <Expression id="myNOTExprFrPLS" xsi:type="content" contentRefId="MainContent" type="MyExpressionTypeNot" />
```

Obrázek 6.3: Instance Výrazu

Do Listu navázání se nezapíše pouze propojení hodnoty prvku s proměnnou, ale je možné propojovat i některé *Vlastnosti* prvku. Jednou z nejpoužívanějších *Vlastností* navázanou na proměnnou ve vizualizaci je *Viditelnost* (Visibility) prvku. Indikátory binárních stavů na stránce Servis jsou založeny na propojení binárního stavu proměnné s *Viditelností* zeleného čtverečku. Pod zeleným čtverečkem je červený čtvereček. Pokud má proměnná hodnotu True, je zelený čtvereček viditelný a červený čtvereček je zakryt. Pokud má hodnotu False, zelený čtvereček vidět není a zobrazí se čtvereček červený.

6.3 Akce a události (Events and actions)

Dalšími důležitými body vizualizace jsou *Akce a události*. Vykoná-li se nějaká událost (např. kliknutí na prvek), spustí se akce. Tyto akce umožňují pracovat s proměnnou (OPC UA akce), měnit *Vlastnosti* prvků (Widget akce), odhlásit nebo přihlásit uživatele (Klient akce) atd. Veškeré události pro určený prvek je možné dohledat v nápovědě. Stejně tak je možné zde najít, jaké akce se dají použít a jaká je jejich syntaxe. Na rozdíl od *Listu navázání*, kam MappView samo zapisuje navázání prvku s proměnnou, musí se *Akce a události* psát ručně do souboru *Navázání události* (Eventbinding) v *Konfiguračním přehledu*. Pokud vývojář chce ve své vizualizaci použít tyto funkce, musí ovládat psaní XML kódu.

7 Rozložení vizualizace (Layout)

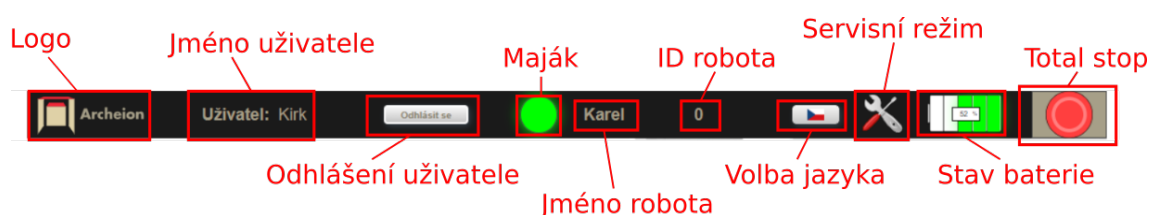
Tato kapitola obsahuje informace o použitých oblastech ve vizualizaci a jejich účel. Dále je zde popsán nový prvek **Servisní režim**², který umožňuje ovládání robota.

Pro snadnější vytváření a nižší výpočetní výkon vizualizace bylo *Rozložení vizualizace* rozděleno na tři oblasti *Hlavička* (AreaHeader), *Hlavní obsah stránky* (AreaContent) a *Navigace* (AreaNavigation) viz Obr. 5.1. Každá stránka má svůj vlastní *Obsah stránky*, který se zobrazuje v *Hlavním obsahu stránky*. *Hlavička* a *Navigace* jsou stejné pro všechny stránky.

Předchozí i nové *Rozložení* mají stejný poměr stran a mají stejné tři oblasti pro *Obsah stránky*. Na některých stránkách v předchozí vizualizaci bylo obtížné rozmístit prvky tak, aby se na stránku vešly. Proto je nové *Rozložení* větší, aby bylo možné umístit na stránku větší počet prvků a lépe se s nimi dalo manipulovat.

7.1 Hlavička (HeaderContent)

Hlavička je umístěna v horní části *Rozložení*. Tato část webových stránek obvykle obsahuje informace, které je dobré mít pořád na očích. Jsou zde umístěny prvky běžné pro téměř každou vizualizaci: logo, jméno přihlášeného uživatele, tlačítko pro odhlášení, možnost změny jazyka. Kromě těchto prvků jsou zde umístěny další potřebné prvky pro identifikaci robota (ID a jméno robota), zobrazení stavu robota pomocí *Majáku*³ a možnost zastavení robota tlačítkem **Total stop** viz Obr. 7.1.



Obrázek 7.1: Hlavička (HeaderContent)

Oproti předchozí verzi vizualizace zde přibýlo tlačítko pro spuštění servisního režimu robota a indikátor stavu nabití baterie. Tento indikátor je inspirován zobrazením stavu nabití a nabíjení baterie z mobilních telefonů.

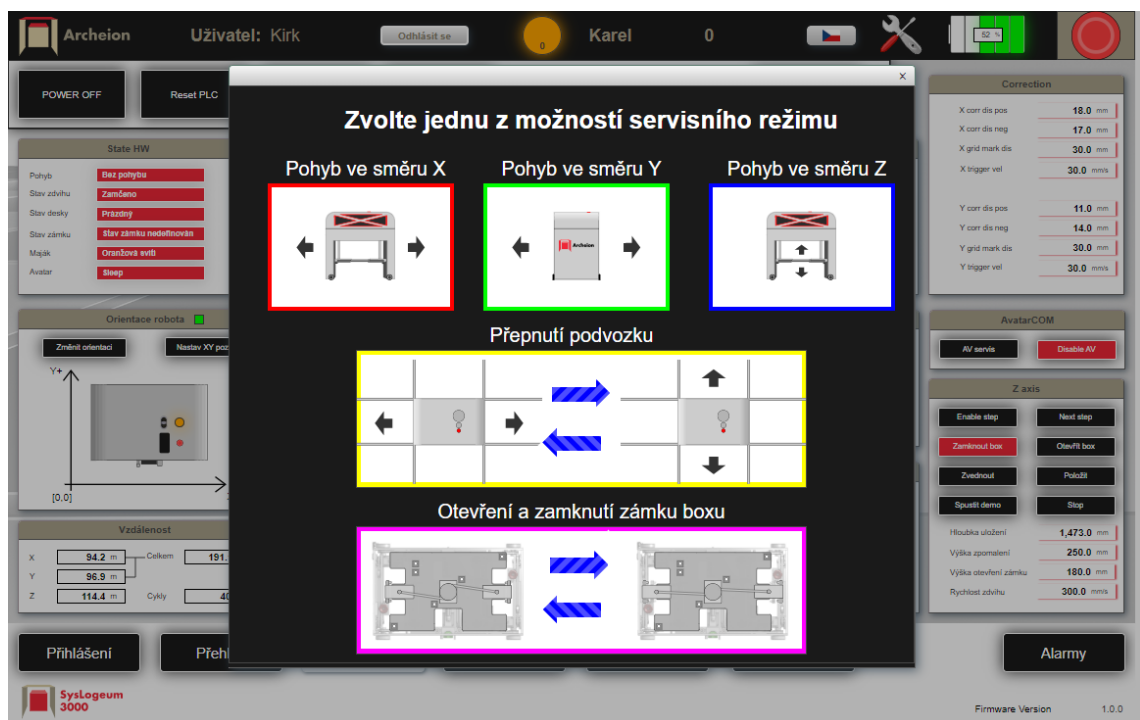
²Názvy prvků, které jsou použity přímo ve vizualizaci, jsou zvýrazněny tučným písmem.

³Maják je světelný indikátor stavu robota umístěný nad krytem robota, který je možné i stisknout. Pokud je Maják zvýrazněn tučným písmem (**Maják**), je tím myšlen prvek ve vizualizaci. Pokud Maják zvýrazněn není, jedná se o reálnou součástku na robotovi.

7.1.1 Servisní ovládání

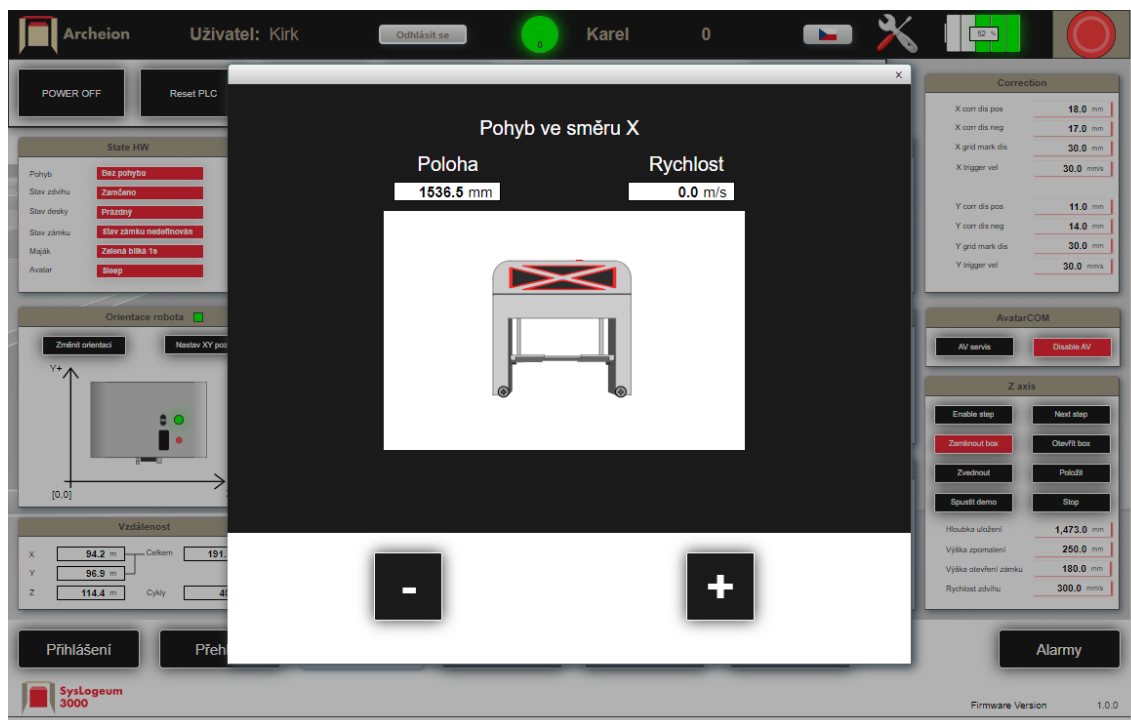
Jedním z největších rozdílů mezi novou a předchozí verzí vizualizace je prvek **Servisní režim** viz Obr. 7.1. Pokud chtěl uživatel v předchozí vizualizaci přepnout robota do servisního režimu, musel držet Maják určitý čas stisknutý. Maják se oranžově rozblíkal a uživatel měl možnost podle počtu stisknutí Majáku zvolit, kterou část robota chce ovládat. Po zvolení mohl tlačítka Plus a Mínus pohybovat danou částí (např. pohyb robota v ose X nebo pohyb desky nahoru a dolů). Pro opuštění servisního režimu uživatel opět musel stisknout Maják po určitou dobu. Uživatel, který neměl k dispozici potřebné údaje o Majáku, tak neměl šanci robota v servisním režimu ovládat.

Z tohoto důvodu byl vytvořen prvek **Servisní režim**. Po stisknutí tohoto prvku se ve vizualizaci otevře dialogové okno. Toto okno obsahuje obrázky jednotlivých částí robota, které je možné ovládat v servisním režimu viz Obr. 7.2. Jednoduchým kliknutím na daný obrázek se otevře další dialogové okno určené pro ovládání dané části viz Obr. 7.3. Pro opuštění servisního režimu stačí pozavírat otevřená okna. Uživatel tak může jednoduše přepínat ovládání jednotlivých částí robota bez potřebné znalosti logiky Majáku. Navíc grafické znázornění je velmi intuitivní.



Obrázek 7.2: Servisní režim (Service mode)

Ačkoliv se ovládání servisního režimu jeví z pohledu vizualizace o dost jednodušší, v pozadí je značně složitější. Veškeré události, které uživatel při ovládání servisního režimu vykoná, spouští v pozadí akce. Tyto akce přepínají robota do režimu, ve kterém se nachází ve vizualizaci. Vizualizace tak přímo zobrazuje způsob, jakým se robot dostává do servisního režimu.



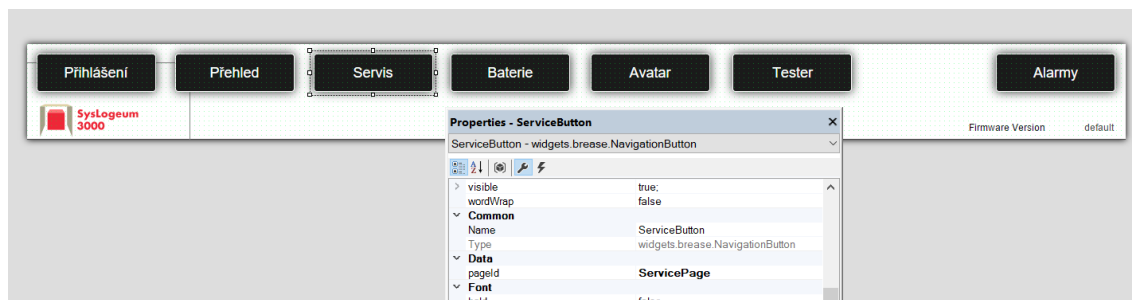
Obrázek 7.3: Servisní ovládání (Service control)

Bylo zapotřebí také vyřešit situaci, kdy dojde při spuštění servisního režimu k náhlému odpojení vizualizace. Po opětovném připojení prohlížeče k vizualizaci se automaticky neobnoví otevřené dialogy. Robot tak zůstane přepnutý v servisním režimu a uživatel jej nemůže ovládat. I tento problém byl vyřešen pomocí *Akcí a událostí*. Pokud je spuštěný servisní režim při opětovném přihlášení uživatele, je mu nabídnuta možnost servisní režim znovu otevřít nebo vypnout.

Možnost ovládat robota jak pomocí Majáku na robotovi, tak pomocí **Majáku** ve vizualizaci však zůstala. Je to pro případ, kdy servisní technik nemá k dispozici vizuální ovládání.

7.2 Navigace (NavigationContent)

Tato část vizualizace obsahuje prvky určené pro přepínání jednotlivých stránek *Navigační tlačítko* (Navigation Button). Po kliknutí na příslušné tlačítko se otevře stránka s ID, které je definováno přímo ve *Vlastnostech* tlačítka viz Obr. 7.4. Mimo tyto prvky obsahuje další logo a aktuální verzi firmwaru robota.



Obrázek 7.4: Navigace (NavigationContent)

8 Návrh vizualizace pro skladovacího robota

Kapitola Návrh vizualizace pro skladovacího robota popisuje, proč byla vizualizace rozdělena na několik stránek, popis těchto stránek a rozdíly oproti předchozí verzi. Jsou zde i popsány další možnosti ovládání robota.

Při návrhu obou vizualizací (předchozí [7] i nové) byly jednotlivé prvky rozmístěny na několik stránek. Je to z toho důvodu, aby vizualizace byla více přehledná a výkonově méně zatížená. Pokud by všechny prvky byly umístěny na jedné stránce, bylo by ovládání robota obtížnější.

Každá stránka slouží k jinému účelu v závislosti na prvcích, které se na ní nacházejí. Podle účelu byl jednotlivým stránkám přiřazen název. Podle názvu stránek byly uživatelům přiřazeny pravomoce pro zobrazení dané stránky.

V nové vizualizaci jsou některé stránky (Přihlášení, Servis, Baterie, Avatar a Alarmy) převzaté z předchozí vizualizace, některé (Logger a Trendy) odstraněné a jiné (Tester a Přehled) přidáné.

Během provozu se ukázalo, že stránka Logger není potřebná. Tato stránka měla smysl pouze pro vývojáře, nikoliv pro uživatele. Stránka Trendy se nepoužívala vůbec. Proto byly tyto stránky odstraněny.

Dalším rozdílem mezi předchozí a novou vizualizací je přidání systému jednotek. Tento systém zobrazuje všechny hodnoty proměnných v základních jednotkách.

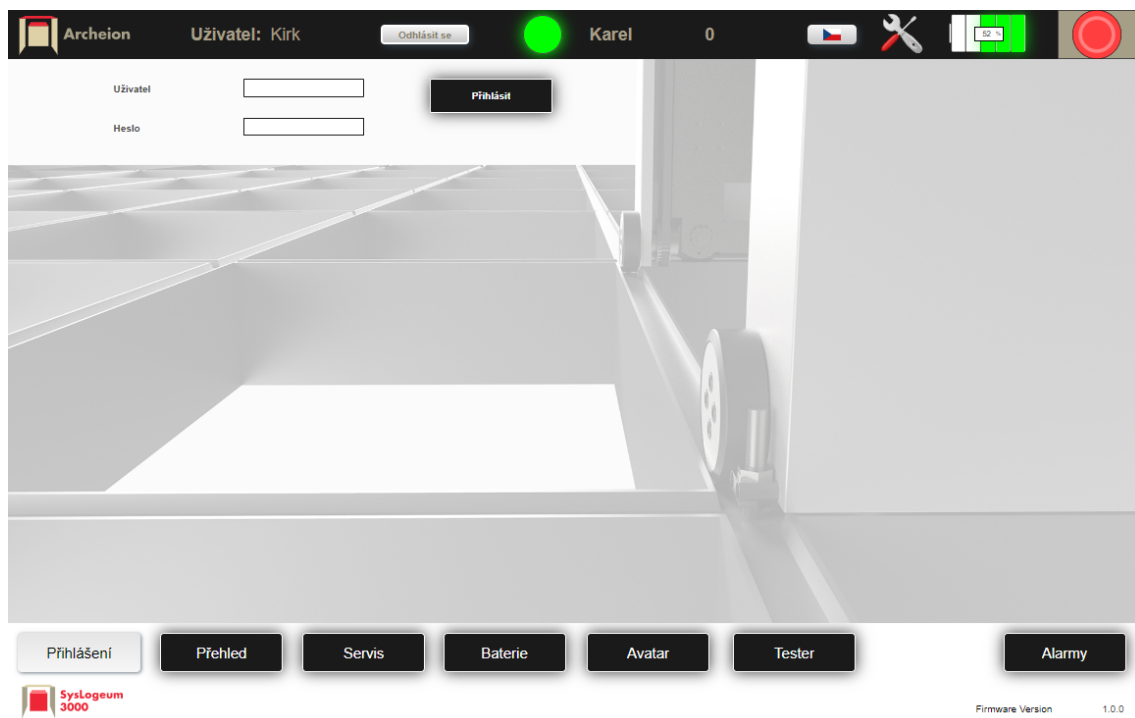
I celkový grafický vzhled nové vizualizace byl přepracován. Jednotlivé části vizualizace byly upraveny podle grafického manuálu firmy Systematic a.s. Celý skladovací systém by tak měl mít jednotný grafický vzhled.

8.1 Přihlášení (Login)

Na stránce Přihlášení se nachází prvek pro přihlášení uživatele. Na tuto stránku je uživatel odkázán po každém připojení k vizualizaci nebo při odhlášení viz Obr. 8.1. Uživatel zde má možnost zadat přihlašovací údaje a stisknout tlačítko **Přihlásit**.

Uživatelům jsou přiřazeny role. Tyto role určují pravomoce každého uživatele viz Obr. 8.2. Pokud není uživatel přihlášen, má roli Kdokoliv (Everyone). Tato role umožňuje uživateli ovládat pouze stránku Přihlášení. K ostatním stránkám má přístup, ale nemá možnost cokoli nastavovat nebo měnit. Po přihlášení je uživateli zpřístupněno ovládání dalších stránek podle jeho pravomocí.

V předchozí verzi byl na této stránce navíc prvek **Změna hesla**. Při provozu vizualizace bylo zjištěno, že tento prvek nefunguje správně. Pokud byli k vizualizaci připojeni dva uživatelé a oba chtěli změnit heslo, navzájem si přepisovali údaje.



Obrázek 8.1: Přihlášení (Login)

Z tohoto důvodu byl tento prvek v nové verzi dočasně odstraněn.

Obrázek v pozadí je použit pro celou vizualizaci. V předchozí vizualizaci bylo pozadí jednobarevné. Tato změna dala vizualizaci modernější vzhled.

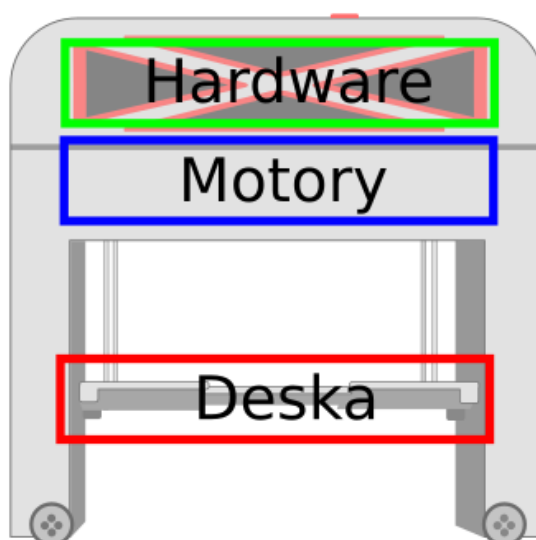
8.2 Přehled (Overview)

Tato stránka je především určena pro uživatele, který má základní znalosti o robotovi, ale nemá přístup k jeho ovládání. Jedná se o stránku, která je zcela nová (v předchozí vizualizaci nebyla). Uživatel zde má možnost sledovat některé parametry robota, podívat se na jednotlivé úrovně konstrukce robota a na jejich jednotlivé komponenty. Je možné zde sledovat pohyb desky s uchopovacím mechanismem při manipulaci s přeprávkou v reálném čase prostřednictvím animace. Pomocí této stránky je uživatel schopen rychle kontrolovat stav HW robota.

Tato stránka je ze všech stránek nejinteraktivnější. Celá je rozdělena na několik částí. Většinu stránky zabírá prvek s kartami. Tyto karty obsahují obrázky jednotlivých konstrukčních úrovní robota (**Hardware**, **Motory**, **Deska**) viz Obr. 8.3 a animaci pohybu desky (**Plošina**). Jednotlivé karty je možné přepínat pomocí tlačítek nad tímto prvkem s kartami.

Role	Práva pro				
	Zobrazení stránky Tester	Zobrazení zbylých stránek	Ovládání stránky Servis	Ovládání stránky Avatar	Ovládání stránky Tester
Everyone	Ne	Ne	Ne	Ne	Ne
Operator	Ne	Ano	Ne	Ne	Ne
Service	Ano	Ano	Ano	Ne	Ano
Avatar	Ne	Ano	Ne	Ano	Ne
Supervisor	Ano	Ano	Ano	Ano	Ano

Obrázek 8.2: Uživatelské role



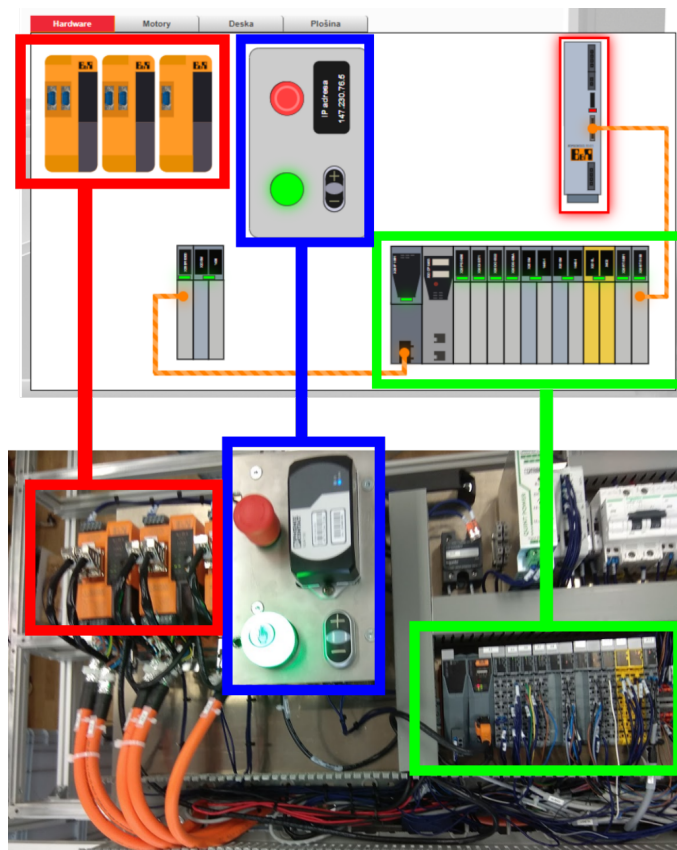
Obrázek 8.3: Konstrukční úrovně robota

8.2.1 Hardware

Na první kartě je úroveň **Hardware**. Jedná se o část robota umístěnou pod krytem. Nachází se zde modulové PLC od firmy B&R, moduly pro řízení motorů, modul pro nabíjení a ovládací panel robota. Všechny prvky jsou rozmístěny obdobně jako na skutečném robotovi viz Obr. 8.10.

Ovládací panel na této stránce je vizuální kopie reálného panelu na robotovi. S vizuální kopií panelu je možné zacházet stejně jako s reálným panelem. Uživatel může stisknout tlačítko **Total stop**, **Maják**, **Plus** a **Mínus** a robot se bude chovat, jako by uživatel ovládal reálný panel. Jedná se tak o další možnost ovládání robota.

Na každý modul na kartě je možné kliknout. Po kliknutí na modul se modul označí modře a přepne se prvek **Vlastnosti** viz Obr. 8.5. V prvku **Vlastnosti** se zobrazí vlastnosti daného modulu (název modulu, navázané proměnné) a prvek **Nápověda**. Po kliknutí na **Nápovědu** se otevře okno obsahující technické informace o daném modulu viz Obr. 8.6. Každý modul firmy B&R obsahuje LED indikátory.



Obrázek 8.4: Porovnání HW robota s vizualizací

Pomocí těchto indikátorů je možné diagnostikovat stav modulu. V případě chyby modulu se daný modul označí červeně (ve vizualizaci i ve skutečnosti). Uživatel tak spolu s vizualizací může jednoduše detekovat chybný modul a možnou příčinu.

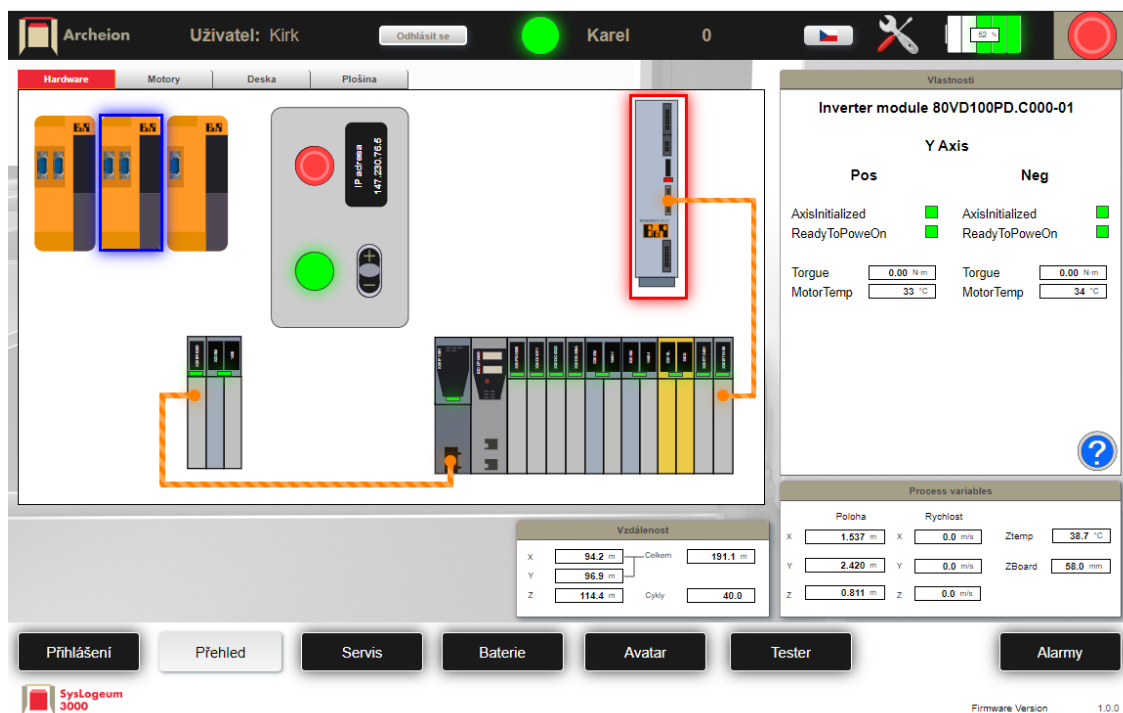
8.2.2 Motory

Pro pohyb ve směru X a Y jsou použity čtyři motory. Pohyb desky je ovládán jedním motorem. Pro přepnutí podvozku ze směru X na směr Y (a naopak) jsou použity další dva motory. Všechny tyto motory je možné vidět na kartě **Motory** viz Obr. 8.7. Vlastnosti motorů jsou obsaženy v modulech, které je řídí. Proto se po kliknutí na motor nic nestane. Pokud by firma Systematic a.s. měla zájem, mohla by i tato stránka být interaktivní (motory a převody by se mohly pohybovat přesně podle robota).

Kromě motorů jsou zde zobrazeny koncové senzory motorů pro přepínání podvozku, senzor na přítomnost přepravky a senzory pro detekce hran gridu.

8.2.3 Deska

Pohyblivá deska s uchopovacím mechanismem je zobrazena na třetí kartě viz Obr. 8.8. Přímo na desce je umístěn krokový motor, který pohybuje mechanismem pro



Obrázek 8.5: Označený modul

uchopení a uvolnění přepravky. Tento motor včetně uchopovacího mechanismu je zobrazen i na kartě **Deska**. Pozici motoru je možné určit podle koncových senzorů zobrazených na **Desce** nebo podle toho, zda je mechanismus uzavřený nebo otevřený viz Obr. 8.9. Jsou zde ještě umístěny senzory pro detekci přítomnosti přepravky a napájení desky z robota.

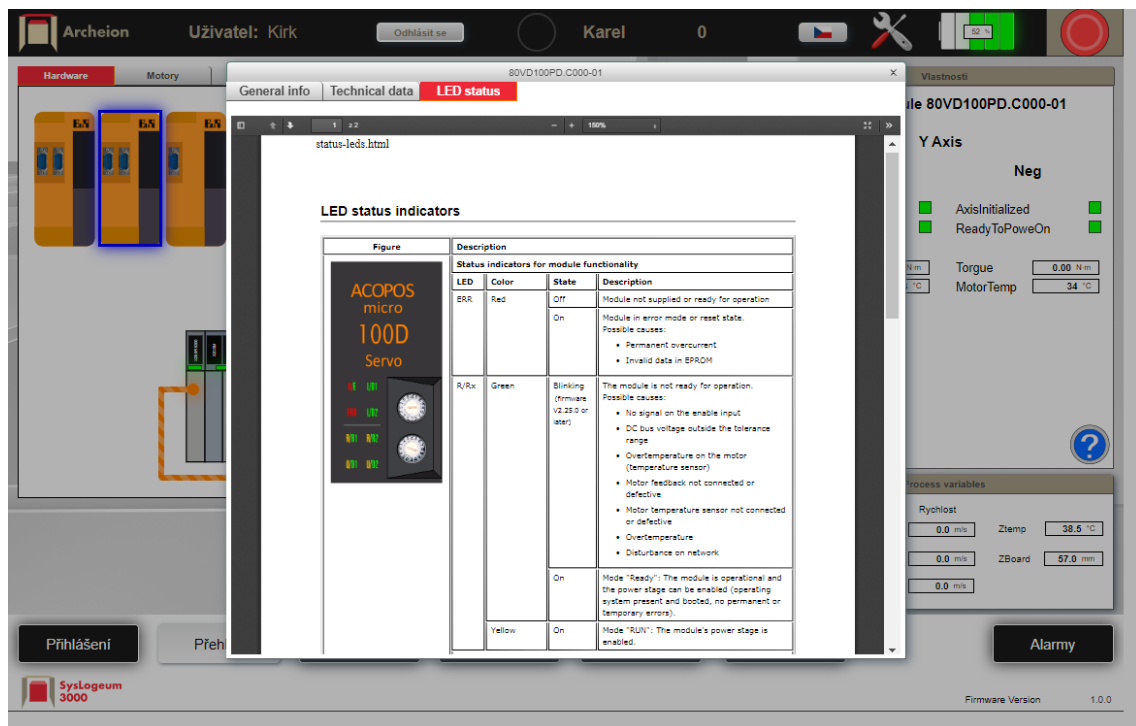
8.2.4 Plošina

Na poslední kartě je umístěna animace pohybu desky při zvedání a pokládání přepravky viz Obr. 8.10. Jedná se o ukázkou, jakým směrem by mohl vývoj vizualizace pokračovat.

Aktuální stav robota je možné zjistit pomocí údajů na stránce. Vzhledem k tomu, že se jedná především o číselné a binární údaje, je i pro zaškoleného uživatele obtížné určit stav robota. Pomocí animací robota, které by zobrazovaly skutečný aktuální stav během provozu, by bylo o dost jednodušší určit stav robota. Tyto animace by mohly sloužit i pro servisního technika k určení příčiny chyby robota. Technik by si mohl prostřednictvím animací přehrát děj, který předcházal chybě.

Vytvořit webovou animaci není složité. Vývojář potřebuje mít základní znalosti o HTML5 a CSS3. Do HTML5 kódu je možné vložit SVG obrázek a odkaz na styl CSS3. V CSS3 kódu se pomocí jednoduchých příkazů dají transformovat (zvětšovat, zmenšovat, rotovat, posouvat) jednotlivé objekty SVG obrázku. [13]

Jak jsem zmínil výše mappView vývojáři neumožňuje přístup k těmto technologiím. Proto existuje prvek *Papír*. Tento prvek je možné napojit na SVG obrázek



Obrázek 8.6: Nápopvěda modulu

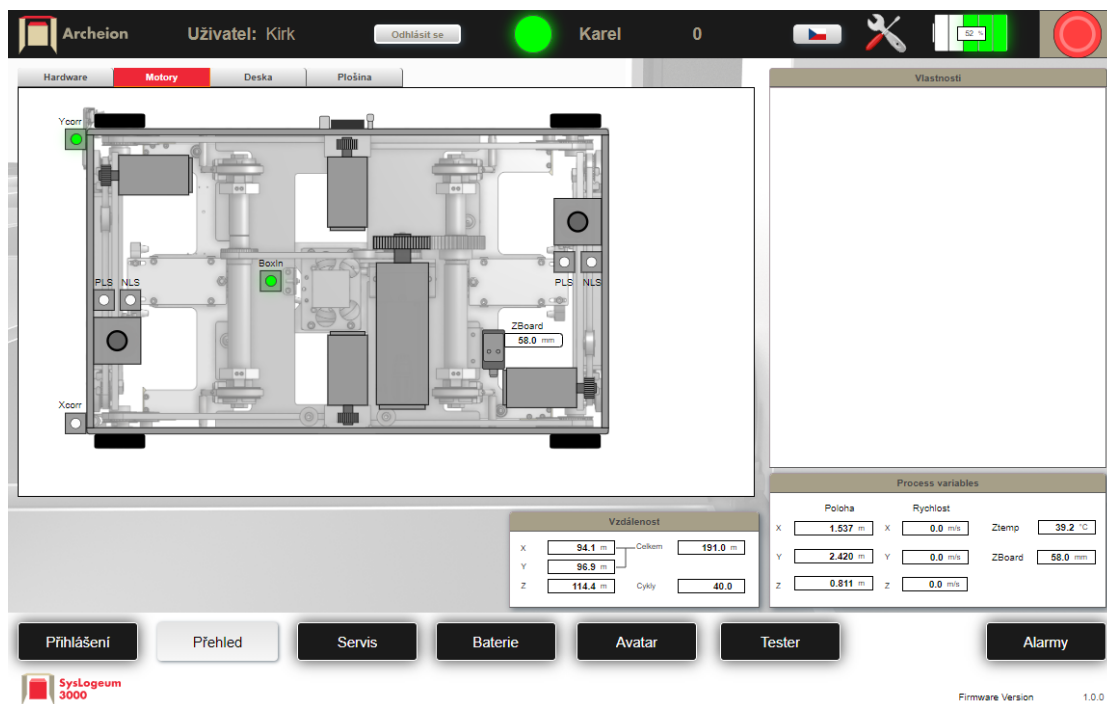
a proměnnou obsahující text (typ String). Do proměnné je možné zapisovat transformace jednotlivých objektů přímo z programu. Pokud vývojář používá stavový automat, je možné v každém stavu transformovat SVG obrázek tak, aby odpovídal skutečnosti.

K tomu, aby bylo možné použít prvek *Papír* je zapotřebí licence, kterou tento projekt nemá. Pro zobrazení animace bylo nutné najít jiné řešení.

Další možností, jak ve vizualizaci zobrazovat animace, je grafický formát GIF. Tento formát je určený pro rastrovou grafiku, ale může obsahovat i jednoduché animace. V mapView je možné jej vložit do prvku *Obrázek* (Image) (pro tento prvek není potřebná licence). Pomocí *Akcí a událostí* je možné měnit obsah prvku *Obrázek*. Pro animaci bylo vytvořeno několik GIFů, které zobrazují jednotlivé stavy pohybu (spuštění desky, uchopení přepravky, zvednutí přepravky atd.). Při reálném pohybu desky jsou v závislosti na stavech automatu spuštěny události. Tyto události vyvolávají akce, které přepínají jednotlivé GIFy zobrazované na *Obrázku*. Vytvořená animace tak vizualizuje aktuální stav pohybu desky. Je tak možné sledovat pohyb desky robota bez nutnosti ji přímo vidět.

8.3 Ovládání robota

Ovládání je možné ze dvou úrovní viz Obr. 8.11. První je lokální úroveň. Tato úroveň je určená pro servisního technika (role Service). Druhá úroveň řízení umožňuje ovládat robota dálkově pomocí struktury Avatar (z nadřazeného systému). Nadřazeným



Obrázek 8.7: Motory

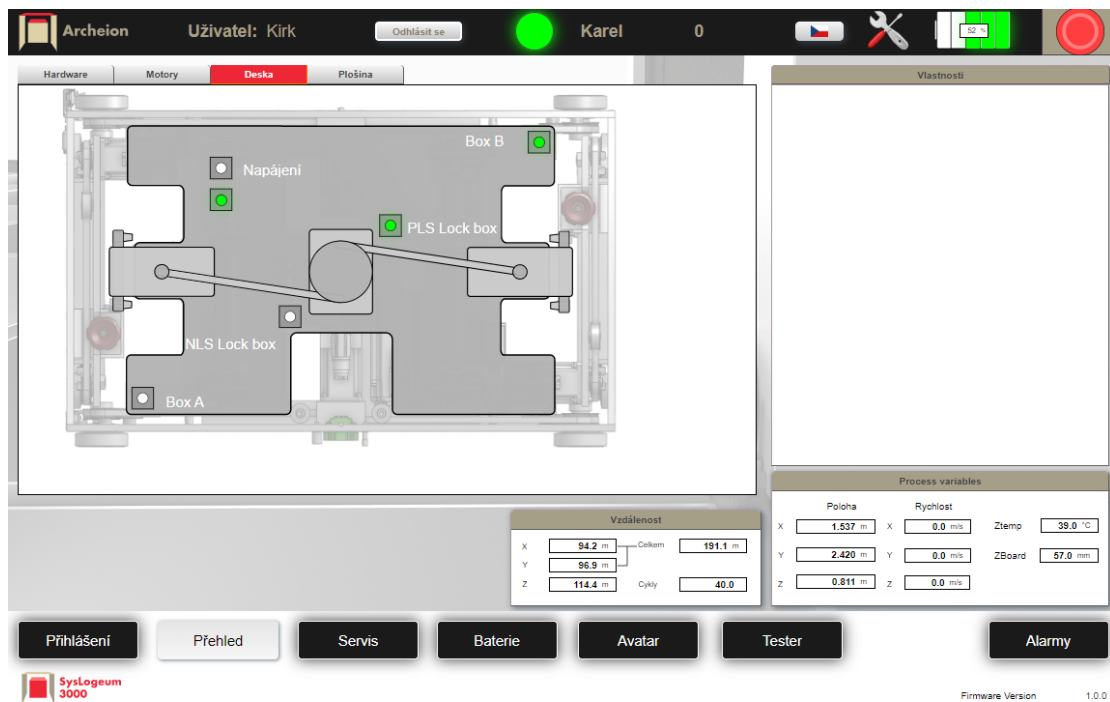
systemem je plánovač. Plánovač vytváří strategii pro přesun přepravek. Z výsledného pohybu vyplývají řídicí příkazy pro jednotlivé roboty předávané přes Avatar. Simulaci plánovače umožňuje stránka Avatar (role Avatar).

Ovládání Avatar slouží k zadávání koncových pozic pohybu robota v souřadném systému XY. Na rozdíl od Avataru umožňuje stránka Servis přímé ovládání pohybu robota v jednotlivých osách.

8.4 Servis (Service)

Stránka Servis vychází z předchozí verze. Veškeré prvky jsou uspořádány do skupin, aby bylo rozpoznatelné, které prvky patří k sobě a k čemu slouží. Pomocí stránky Servis může servisní technik vypnout ovládání robota z nadřazeného systému a ovládat robota. Technik zde má několik možností pro ovládání robota:

- Přepínat robota na směr X nebo Y a pohybovat v daném směru s ním.
- Uzamknout podvozek (díky tomu není možné s robotem pohnout ve směru X ani Y) a pohybovat deskou, popř. zvednout nebo položit přepravku.
- Vyčítat a potvrzovat chyby a varování.
- Po dosažení pozice u nabíječky spustit nabíjení robota.
- Restartovat PLC.



Obrázek 8.8: Deska

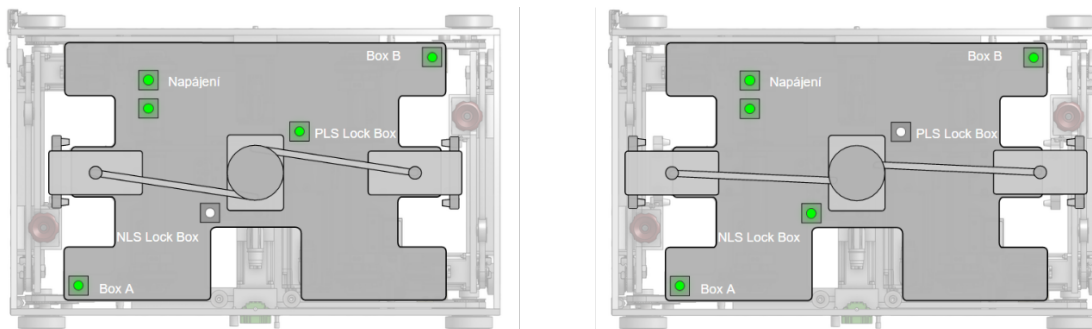
- Sledovat stavové proměnné a binární stavy robota.
- Nastavovat korekce robota.

Korekce robota slouží k přesnému dosažení požadované pozice.

Všechny tyto možnosti byly i v předchozí verzi a jsou nezbytné pro servisní ovládání robota. Přibyly zde ale i další prvky, které v předchozí verzi nebyly. Jedná se o možnosti:

- Vypnout robota.
- Smazat všechny chyby nebo varování.
- Nastavit orientaci robota.

Při nasazení robota na grid je k úspěšnému ovládání potřeba definovat orientaci robota. Způsob, jakým je určena orientace robota, je poloha nabíjecího konektoru. Robota je možné usadit buď v záporném směru Y (nabíjecí konektor směřuje do záporného směru osy Y), nebo v kladném směru Y (nabíjecí konektor směřuje do kladného směru osy Y). To, jakým způsobem bude robot usazen, se může v každém skladu lišit. Proto na této stránce přibyl prvek **Orientace robota**. Po usazení robota na grid se servisní technik připojí k vizualizaci. Pokud obrázek s pozicí konektoru ve vizualizaci nesouhlasí s pozicí konektoru ve skutečnosti, stačí technikovi stisknout tlačítko **Změna orientace**. Tím se změní obrázek viz Obr. 8.12 a i souřadný systém pro robota (prohodí se kladný a záporný směr osy X a i osy Y). Na obrázku je



Obrázek 8.9: Uzavřená/otevřená deska

možné vidět **Maják**. Opět se jedná o vizuální kopii Majáku na robotovi. Obrázek tak vypadá více interaktivní.

To, co na této stránce není a bylo v předchozí verzi, je **Maják** a tlačítka **Plus** a **Mínus** v prostřední části stránky. Tyto prvky byly nahrazeny již zmíněným novým prvkem **Servisní režim**. Přílohy této práce obsahují informace potřebné k ovládání robota pomocí **Majáku** a tlačítek **Plus** a **Mínus**. Tyto informace ukazují, jak bylo obtížné ovládat robota tímto způsobem.

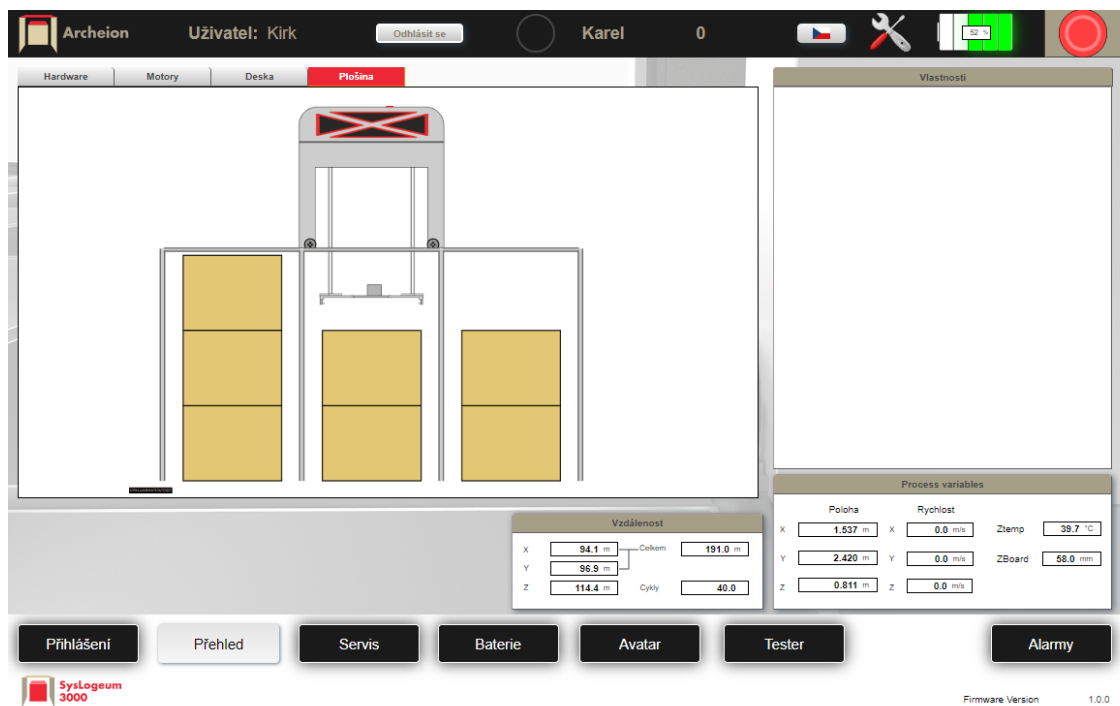
8.5 Avatar

Po povolení ovládání Avataru ze stránky Servis může uživatel s rolí Avatar využít veškeré prvky na stránce Avatar. Tato stránka slouží jako simulace nadřazeného systému. Robot se chová tak, jako kdyby dostával příkazy od nadřazeného systému. Uživatel zde může robotem pohybovat ve směru X nebo Y, zvedat nebo pokládat přepravku, spustit a zastavit nabíjení a potvrzovat chyby a varování.

Předchozí verze této stránky je téměř stejná. V ideálním případě by se komunikace mezi robotem a nadřazeným systémem neměla měnit. Proto není potřeba provádět velké úpravy na této stránce. I tak zde přibyly tlačítka **Plus** a **Mínus** pro jednodušší posouvání robota. Po stisknutí daného tlačítka se automaticky nastaví hodnota další pozice v daném směru. Uživatel tak nemusí po každé zadávat požadované souřadnice, jako to bylo v předchozí verzi. Dále ještě přibyly prvky s údaji o baterii.

8.6 Baterie (Battery)

Stránka Baterie slouží k poskytování informací o bateriích robota. Tato stránka je čistě informativní (nelze zde nastavovat hodnoty). Předchozí verze robota měla vlastní správu pro baterie, jejíž činnost se zobrazovala na této stránce. Během provozu se zjistilo, že tato správa není nezbytná. Proto byla odstraněna, a tím se i snížil počet prvků na této stránce. I tak se na první pohled může zdát, že obsah této stránky je chudý. To by však nemělo být na dlouho. Do budoucna se plánuje vytvoření systému, který bude zaznamenávat údaje o baterii při každém nabití.



Obrázek 8.10: Plošina

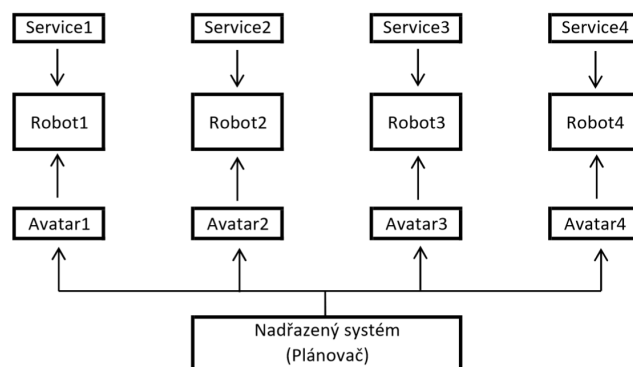
Z dlouhodobého hlediska by pak mělo být možné zjistit úroveň opotřebení baterie a včas jí vyměnit. Volný prostor na této stránce je určen pro tyto údaje.

8.7 Alarmy (Alarms)

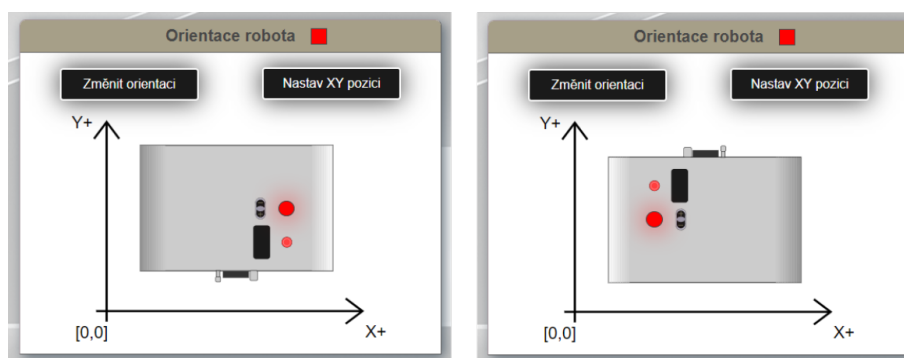
Všechny chyby a varování jsou zobrazovány na stránce Alarmy. Pomocí *Listu alarmů* (AlarmList) a *Historie alarmů* (AlarmHistory) je možné zobrazovat nejen chyby a varování, ale i jejich vlastnosti (datum, vážnost) a stav. AS disponuje řadou mapp komponent. Každá z těchto komponent má svůj daný účel. Vývojář může jakoukoliv z nich přidat do projektu a konfigurovat. Jednou z těchto mapp komponent je *mpAlarmX*. Tato komponenta je určena ke správě chyb a varování. Pomocí softwarového spojení *mpLink* je možné tuto komponentu propojit s *Listem alarmů* a *Historií alarmů*. *List alarmů* pak zobrazuje aktuálně aktivní chyby a varování.

Nad tímto prvkem jsou umístěna dvě tlačítka **Potvrdit** a **Potvrdit vše** sloužící pro potvrzení chyby nebo varování. Pokud je daná chyba nebo varování potvrzena a je již neaktivní, sama je smazána z *Listu alarmů*. Pokud ne, je daná chyba zobrazena, dokud nejsou tyto dvě podmínky splněny. *Historie alarmů* zobrazuje všechny změny stavu chyb a varování (kdy byla daná chyba nebo varování aktivní, kdy neaktivní a kdy byla potvrzena). K oběma prvkům je přidáno tlačítko **Filtr**, které umožňuje filtrovat zobrazené chyby a varování.

Tato stránka se jako jediná oproti předchozí verzi nezměnila.



Obrázek 8.11: Úrovně ovládání



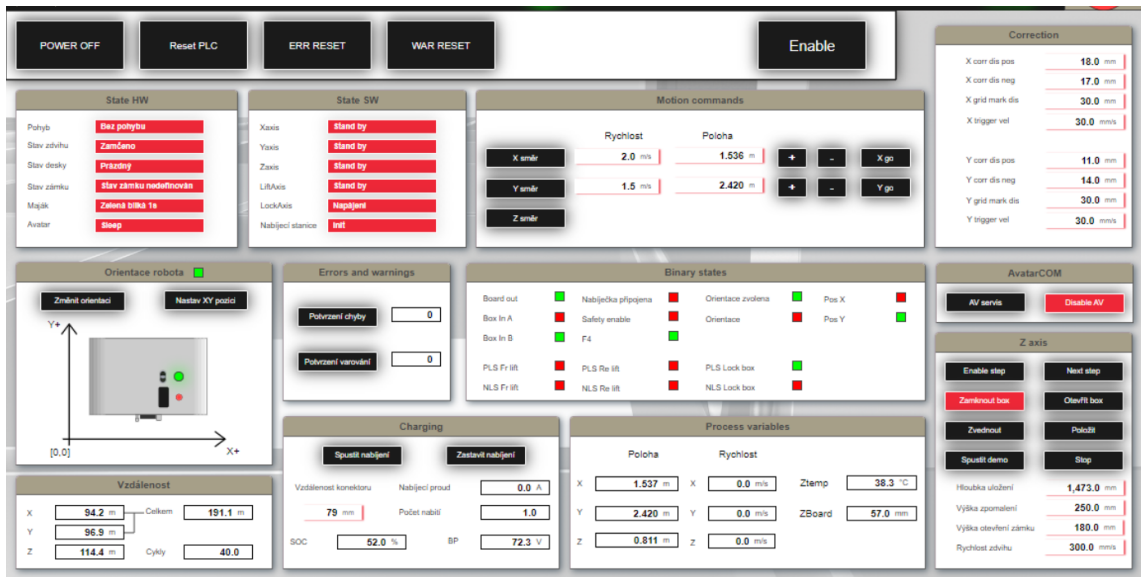
Obrázek 8.12: Orientace robota

8.8 Tester

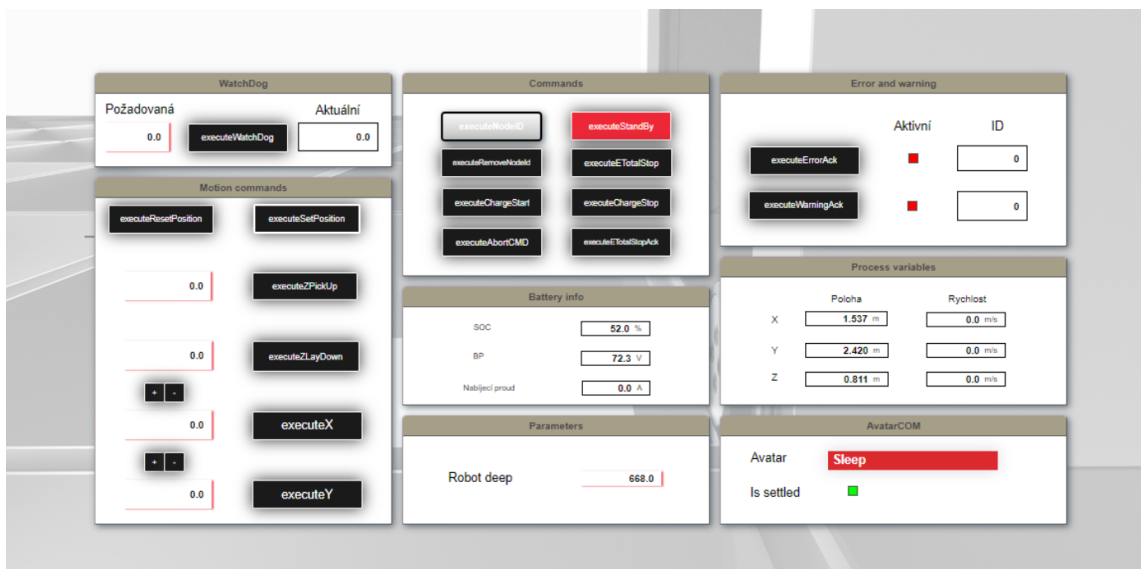
Poslední stránkou vizualizace je stránka Tester. Pomocí této stránky je možné zadat robotovi jednotlivé úkony, které pak bude robot vykonávat sekvenčně (např. pohyb ve směru X -> zvednutí přepravky -> pohyb ve směru Y -> pohyb ve směru X -> položení přepravky). Vývojář musí jednotlivé úkony napsat do textového souboru v programovacím jazyce vytvořeném speciálně pro tuto vizualizaci (tento jazyk je podobný G-kódu). Z vizualizace je pak možné tento kód nahrát a použít pro ovládání robota. Jedná se tak o další možnost ovládání robota.

Výhodou tohoto ovládání je, že robot může sekvenci vykonávat pořád dokola. Při vývoji nové verze robota je tak možné vytvořit simulaci plného provozu bez potřeby nadřazeného systému.

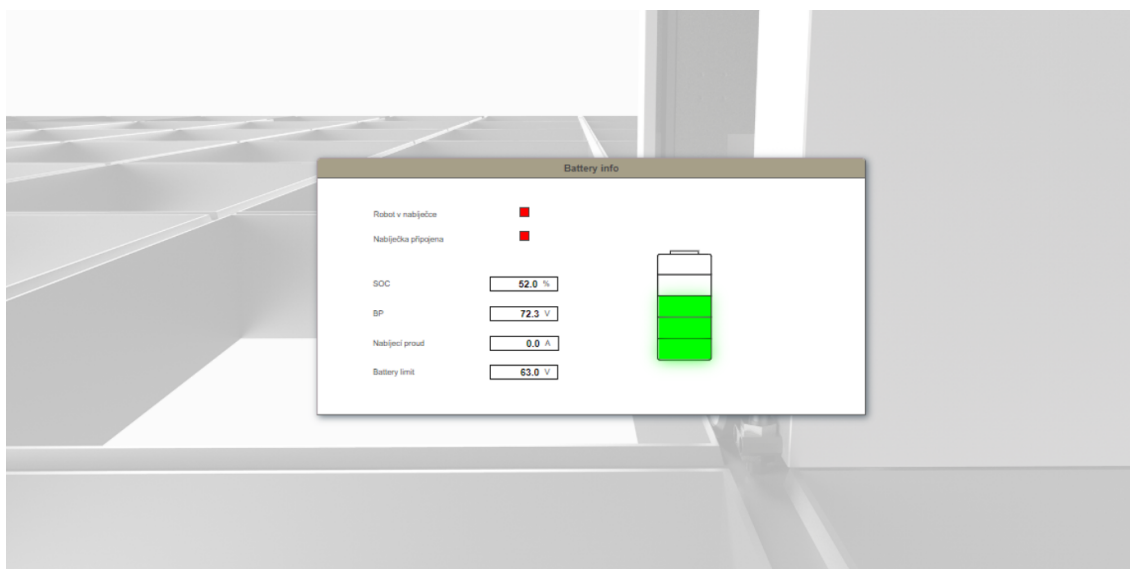
Celá tato stránka (včetně vytvořeného programovacího jazyka a úprav programu robota) je diplomovou prací studenta Bc. Jiřího Kulicha. [14]



Obrázek 8.13: Servisní ovládání



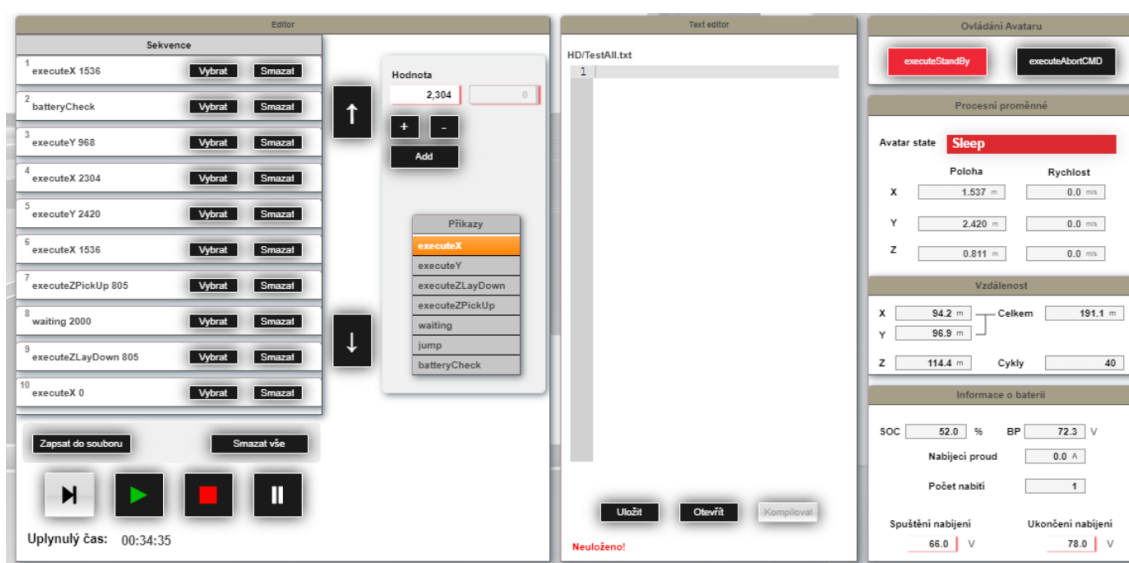
Obrázek 8.14: Stránka pro ovládání ze systému Avatar



Obrázek 8.15: Baterie

Alarmy a chyby			Historie		
Datum a čas	Zpráva	Stav	Datum a čas	Zpráva	Stav
23. dubna 2020 14:34:19	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)		23. dubna 2020 14:34:19	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)	
23. dubna 2020 14:34:19	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)		23. dubna 2020 14:34:19	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	
23. dubna 2020 14:20:56	Chyba v Zaxis 4022: Převrácení se při vytahování nepohybuje shodnou rychlostí jako zvedací plošina		23. dubna 2020 14:20:56	Chyba v Zaxis 4022: Převrácení se při vytahování nepohybuje shodnou rychlostí jako zvedací plošina	
			23. dubna 2020 13:52:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	
			23. dubna 2020 13:33:23	Chyba v Xaxis 7002: Chyba pohonu Xneg	
			23. dubna 2020 13:29:51	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	
			23. dubna 2020 13:21:52	Chyba v LiftAxis: Chyba pohonu RL (RearLift)	
			23. dubna 2020 13:10:34	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)	
			23. dubna 2020 13:05:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	
			23. dubna 2020 13:04:54	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	
			23. dubna 2020 12:59:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	

Obrázek 8.16: AlarmContent



Obrázek 8.17: Stránka Tester pro servisní testování

9 Závěr

Všechny body zadání bakalářské práce byly splněna. Výsledkem této práce je webová vizualizace, která je v současné době nasazena do poloprovozního skladu. Vizualizace spolu s novou verzí robota bude testována provozovatelem skladu. Během testování bude práce ještě upravována tak, aby zcela vyhovovala požadavkům provozovatele. Po dokončení testování se vizualizace stane součástí skladového systému SysLogeum 3000.

Vytvořená vizualizace slouží především servisnímu technikovi a umožňuje mu vzdálenou správu a ovládání robota ve skladu.

Technologie mappView je poměrně nová technologie, která je pořád ve vývoji. Při vytváření některých funkcionalit vizualizace jsem často narazil na omezené možnosti této technologie. V některých případech bylo potřebné vymyslet jiný způsob řešení, který by technologie mappView umožňovala provést.

Nová verze robota včetně nové vizualizace je již 6 týdnů používána v poloprovozním skladu firmy Systematic a.s. Během této doby byly objeveny možnosti nové vizualizace a intuitivnější vzhled. Jako nejvíce přínosný prvek se ukázal nový systém ovládání robota v servisním režimu, který byl uživateli skladu přijat velmi pozitivně.

Jelikož při vývoji nového firmwaru bylo potřebné i robota testovat, bylo nutné, co nejdříve umožnit ovládání robota prostřednictvím vizualizace. Až po té jsem se mohl plně zabývat grafickým vzhledem a přidáváním dalších funkcionalit do vizualizace.

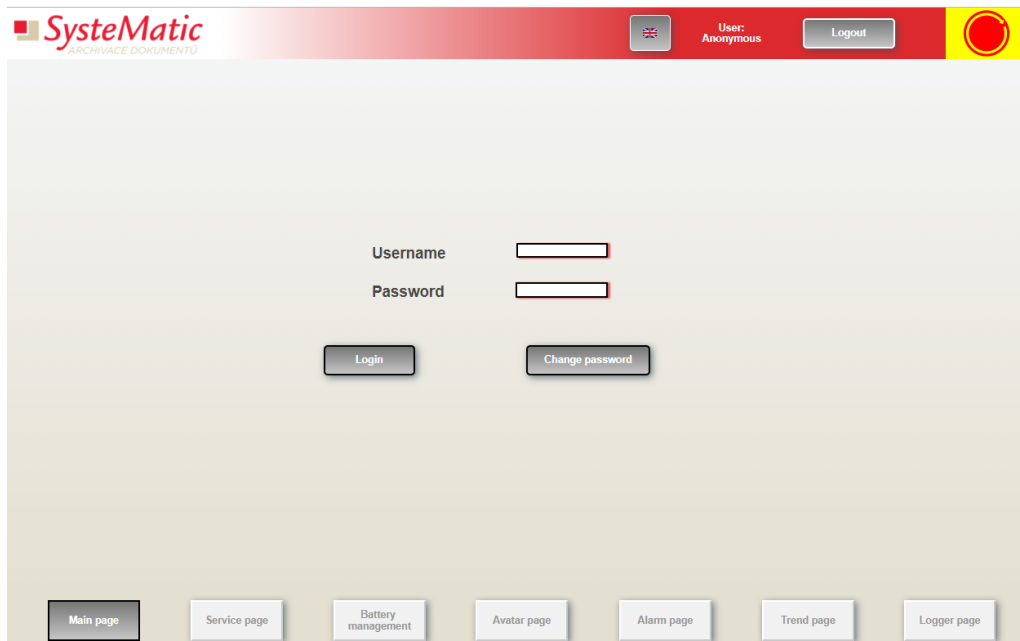
Postupný vývoj technologie mappView otevírá další možnosti využití této technologie pro webové vizualizace. Díky těmto novým možnostem je možné vizualizaci dál zdokonalovat jak po stránce grafické, tak po stránce intuitivity použití. Jedním ze zdokonalení může být implementace prvku *Papír*. Implementací tohoto prvku se stane vizualizace interaktivnější a pro uživatele přívětivější. To vše bude možné jen v případě, že firma Systematic a.s. bude mít zájem o pokračování ve vývoji vizualizace.

Literatura

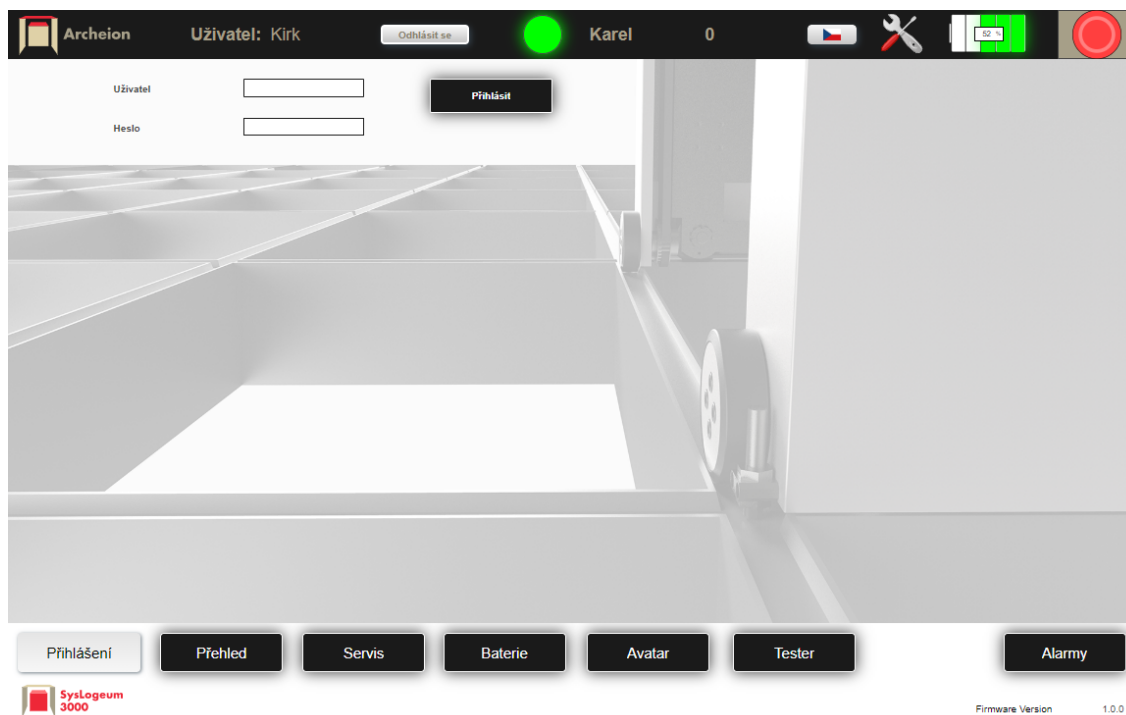
- [1] Goto10 s.r.o. 01_.jpg. In: SKLADOVÝ SYSTÉM ARCHEION [online]. Praha: Systematic [cit. 2020-05-29]. Dostupné z: <https://www.systematic.cz/archeion.htm>
- [2] Benefits. AutoStore [online]. Nedre Vats: AutoStore, 2018 [cit. 2020-05-25]. Dostupné z: <https://autostoresystem.com/benefits>
- [3] Welcome to the automated warehouse of the future. In: The Verge [online]. Washington, D.C.: Vincent, 2018 [cit. 2020-05-10]. Dostupné z: <https://www.theverge.com/2018/5/8/17331250/automated-warehouses-jobs-ocado-andover-amazon>
- [4] What is HMI? Inductive Automation [online]. Folsom (California): Inductive Automation, 2018 [cit. 2020-05-04]. Dostupné z: <https://www.inductiveautomation.com/resources/article/what-is-hmi>
- [5] TM600 – Introduction to Visualization. Eggelsberg: B&R, 2016.
- [6] TM611 – Working with mapp View. Eggelsberg: B&R, 2018
- [7] KUBÍČEK, Tomáš. Návrh HMI pro autonomního skladového robota. Liberec, 2019. Bakalářský projekt. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií. Vedoucí práce Ing. Leoš Beran, Ph.D.
- [8] Scalable Vector Graphics. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-05-25]. Dostupné z: https://cs.wikipedia.org/wiki/Scalable_Vector_Graphics
- [9] BITTNER, Honza. Lekce 4 - Popis vybraných technologií 1. Itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Praha: David Čápka, 2015 [cit. 2020-05-26]. Dostupné z: <https://www.itnetwork.cz/html-css/ostatni/webove-aplikace/technologie-pro-vyvoj-webovych-aplikaci-popis-vybranych-technologiei-1>
- [10] Hypertext Transfer Protocol. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2019 [cit. 2020-05-27]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol

- [11] Alphard. Sessions. In: Péhápkó: Učebnice PHP [online]. Michal Mencl [cit. 2020-05-27]. Dostupné z: <http://www.pehapko.cz/programujeme-v-php/sessions>
- [12] 2. Session proměnné. In: VŠE O WWW [online]. Praha: Jiří Kosek, 2000 [cit. 2020-05-27]. Dostupné z: <https://www.kosek.cz/clanky/php4/session.html>
- [13] Dev Ed. Learn To Build An SVG Animation With CSS. In: Youtube [online]. Zveřejněno 23. 11. 2019 [vid. 2020-5-20]. Dostupné z: <https://youtu.be/gWai7fYp9PY>
- [14] KULICH, Jiří. Editor testovacích tras pro systém SysLogeum 3000. Liberec, 2020. Diplomová práce. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií. Vedoucí práce Ing. Leoš Beran, Ph.D.
- [15] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. 1. díl. Praha: BEN, 1999. ISBN 80-8605- 58-9
- [16] KARL-HEINZ, John a Michael TIEGELKAMP. IEC 61131-3: programming industrial automation systems: concepts and programming languages, requirements for programming systems, aids to decision-making tools. Berlin: Springer, c2001. ISBN 3-540-67752-6.
- [17] Download SinuTrain for SINUMERIK Operate V4.8 - Basic. Siemens [online]. [cit. 2018-10-09]. Dostupné z: https://www.industry.siemens.com/topics/global/en/cnc4you/cnc_downloads/sinutrain_downloads/Pages/download-sinutrain-for-sinumerik-operate-v48-basic.aspx
- [18] EISING, D. V. (2013). Effective console operator HMI design: ASM consortium guidelines. Houston, TX, Abnormal Situation Management (ASM).
- [19] TIŠNOVSKÝ, Pavel. Vektorový grafický formát SVG. In: Root.cz [online]. Praha: Internet Info, 2007 [cit. 2020-05-27]. Dostupné z: <https://www.root.cz/clanky/vektorovy-graficky-format-svg/>
- [20] WULFRAAT, Marc. The Autostore Semi-automated Storage System. In: MW-PVL [online]. Montreal: MWPVL, 2020 [cit. 2020-05-29]. Dostupné z: https://mwpvl.com/html/autostore_review.html
- [21] TM641 – Display alarms, charts and data in mapp View. Eggelsberg: B&R, 2018.
- [22] TM671 – Creating powerful mapp View visualizations. Eggelsberg: B&R, 2018.
- [23] B1. AutoStore [online]. Nedre Vats: AutoStore, 2018 [cit. 2020-05-25]. Dostupné z: <https://autostoresystem.com/b1/>
- [24] Extensible Markup Language. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-06-01]. Dostupné z: https://cs.wikipedia.org/wiki/Extensible_Markup_Language

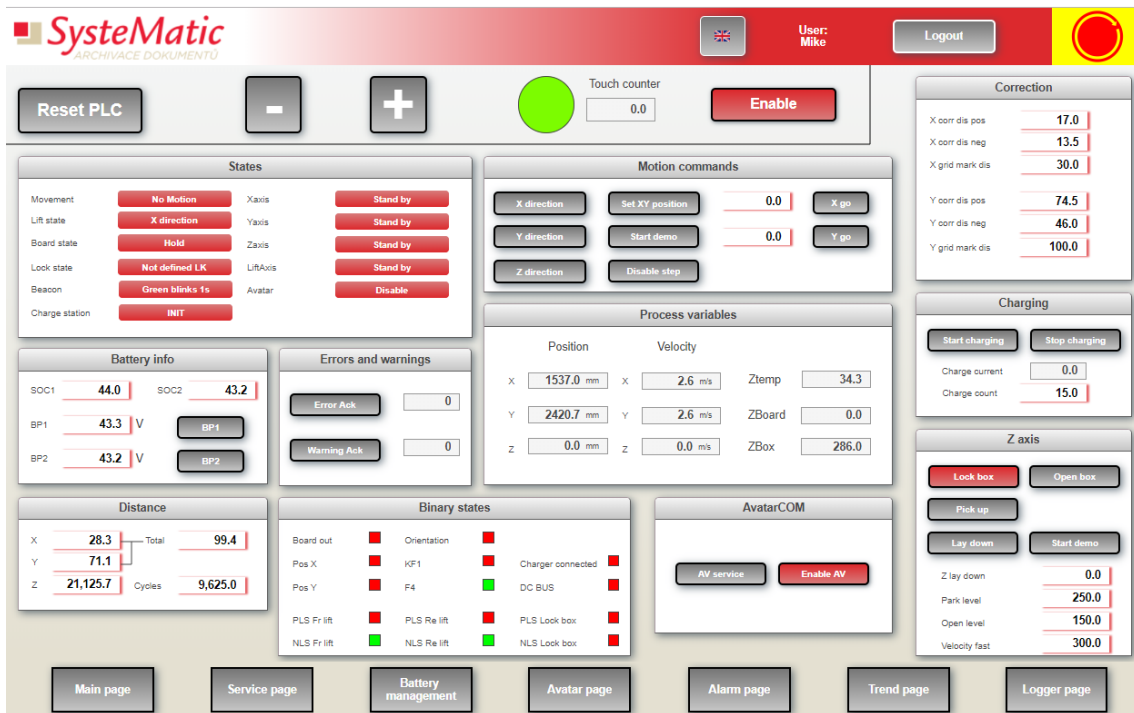
A Porovnání předchozí a nové vizualizace



Obrázek A.1: Stránka Přihlášení (předchozí verze)



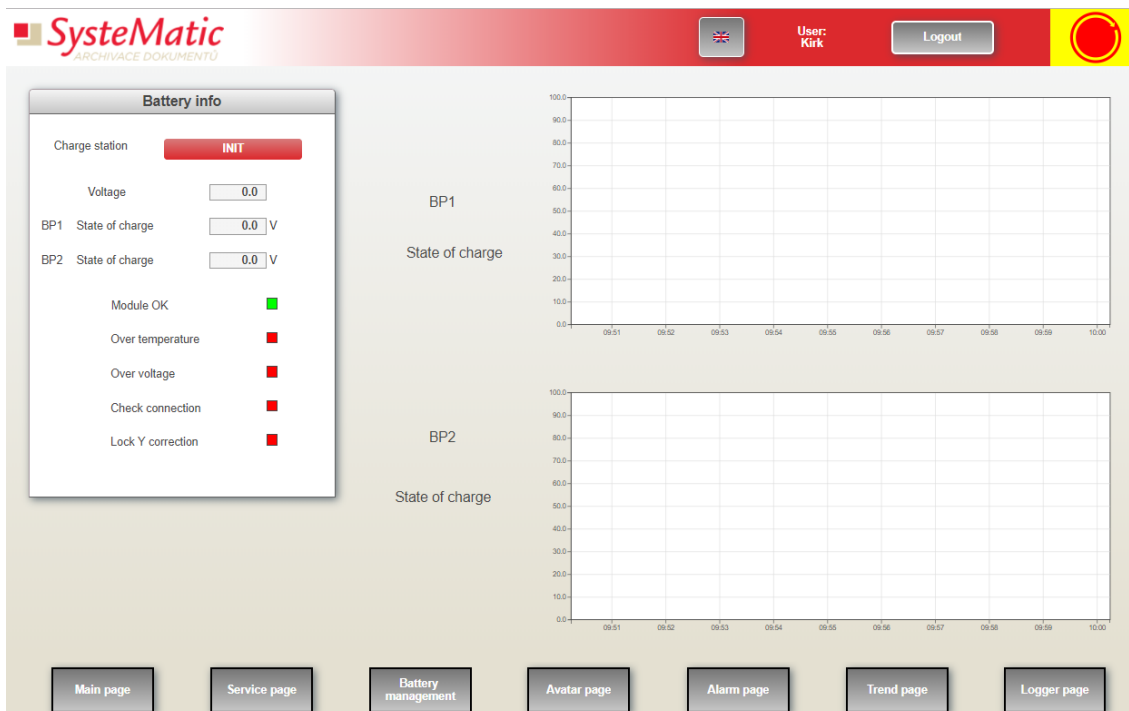
Obrázek A.2: Stránka Přihlášení (nová verze)



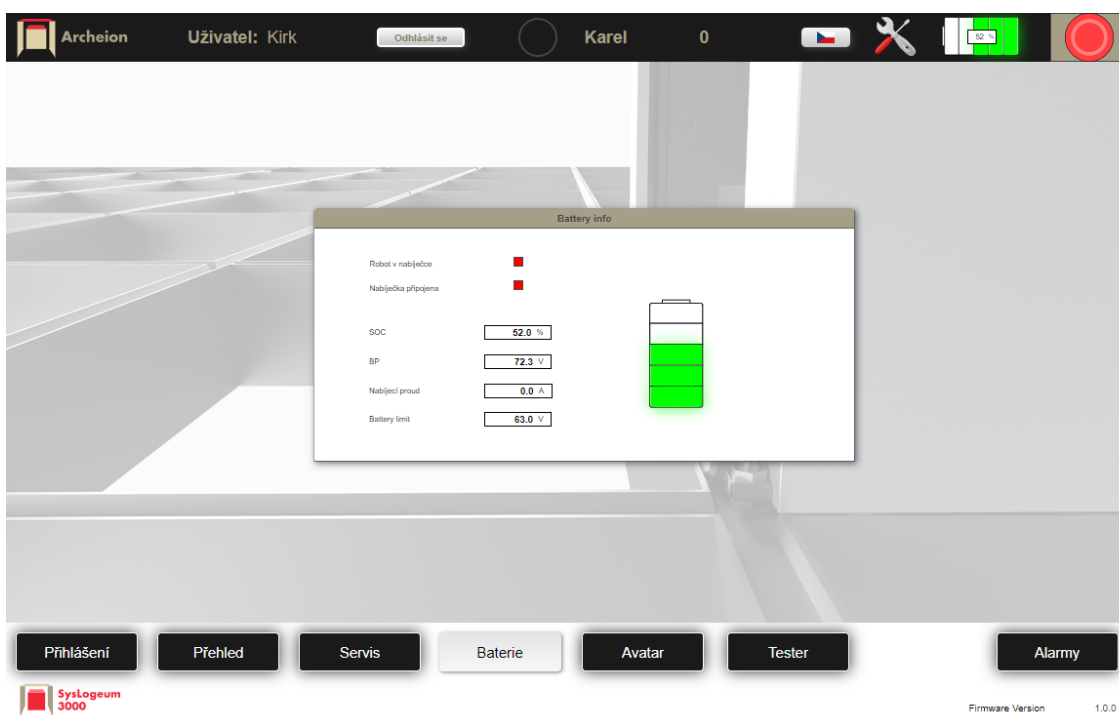
Obrázek A.3: Stránka Servis (předchozí verze)



Obrázek A.4: Stránka Servis (nová verze)



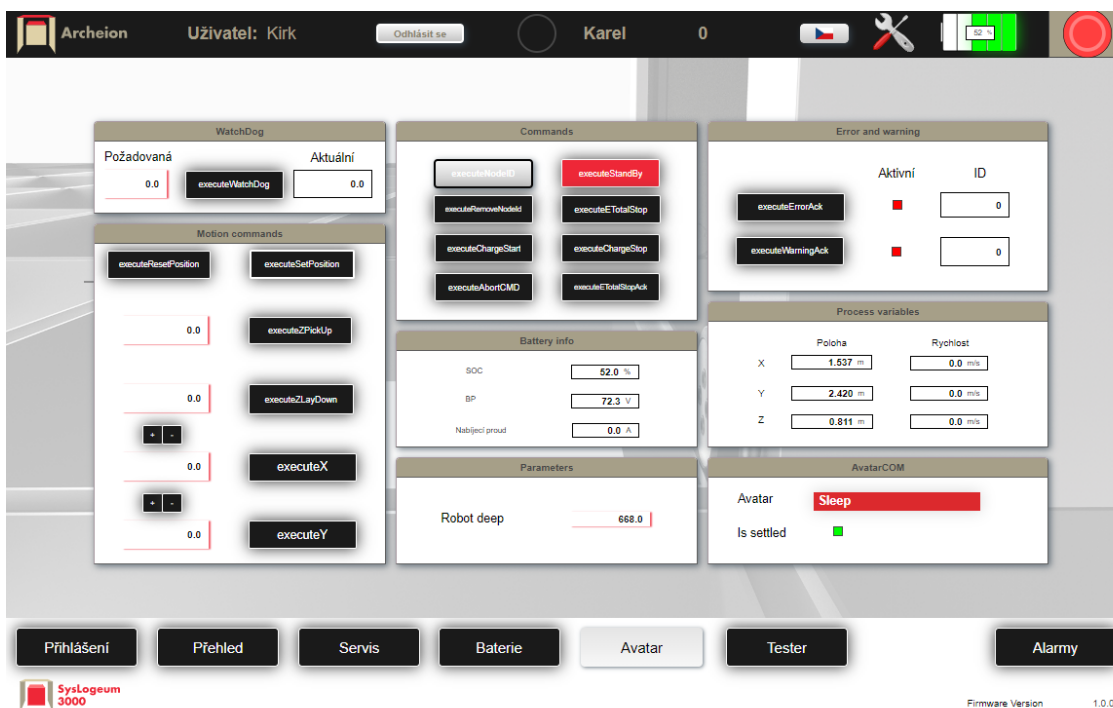
Obrázek A.5: Stránka Baterie (předchozí verze)



Obrázek A.6: Stránka Baterie (nová verze)



Obrázek A.7: Stránka Avatar (předchozí verze)



Obrázek A.8: Stránka Avatar (nová verze)

Systematic
ARCHIVACE DOKUMENTŮ

User: Kirk Logout

Alarms and errors **History**

Date and time	Message	State	Severity
Sunday, May 12, 2019 9:41:50 PM	Warning in BMS_1 5027: chyba ve funkčním bloku knihovny CAN_Lib	⚠	5
Sunday, May 12, 2019 9:41:50 PM	Warning in BMSman: otevření CAN komunikace se nezdařilo	⚠	5
Sunday, May 12, 2019 9:41:50 PM	Error in LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	10
Sunday, May 12, 2019 9:41:50 PM	Error in LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	10

Date and time	Message	Old state	New state	Severity
Sunday, May 12, 2019 3:15:52 PM	Error in LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠	10
Saturday, May 11, 2019 11:04:34 PM	Error in LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠	10

Main page Service page Battery management Avatar page Alarm page Trend page Logger page

Obrázek A.9: Stránka Alarmy (předchozí verze)

Archeion Uživatel: Kirk Odláskit se Karel 0

Alarmy a chyby **Historie**

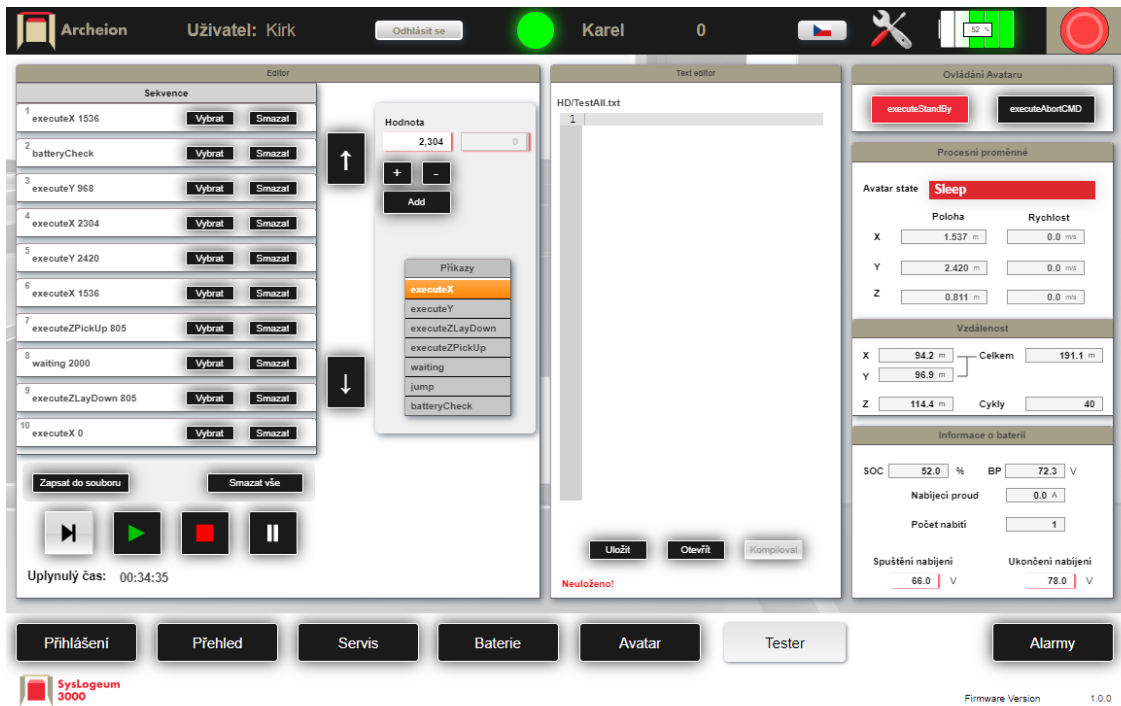
Datum a čas	Zpráva	Stav
23. dubna 2020 14:34:19	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)	⚠
23. dubna 2020 14:34:19	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠
23. dubna 2020 14:20:56	Chyba v Zaxis 4022: Přepravka se při vytahování nepohybuje shodnou rychlostí jako zvedací plošina	⚠

Datum a čas	Zpráva	Starý	Nový
23. dubna 2020 14:34:19	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)	⚠	⚠
23. dubna 2020 14:34:19	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 14:20:56	Chyba v Zaxis 4022: Přepravka se při vytahování nepohybuje shodnou rychlostí jako zvedací plošina	⚠	⚠
23. dubna 2020 13:52:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 13:33:23	Chyba v Xaxis 7002: Chyba pohonu Xneg	⚠	⚠
23. dubna 2020 13:29:51	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 13:21:52	Chyba v LiftAxis: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 13:10:34	Chyba v LKaxis 7008: Chyba pohonu LB (LockBox)	⚠	⚠
23. dubna 2020 13:05:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 13:04:54	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠
23. dubna 2020 12:59:25	Chyba v LiftAxis 7007: Chyba pohonu RL (RearLift)	⚠	⚠

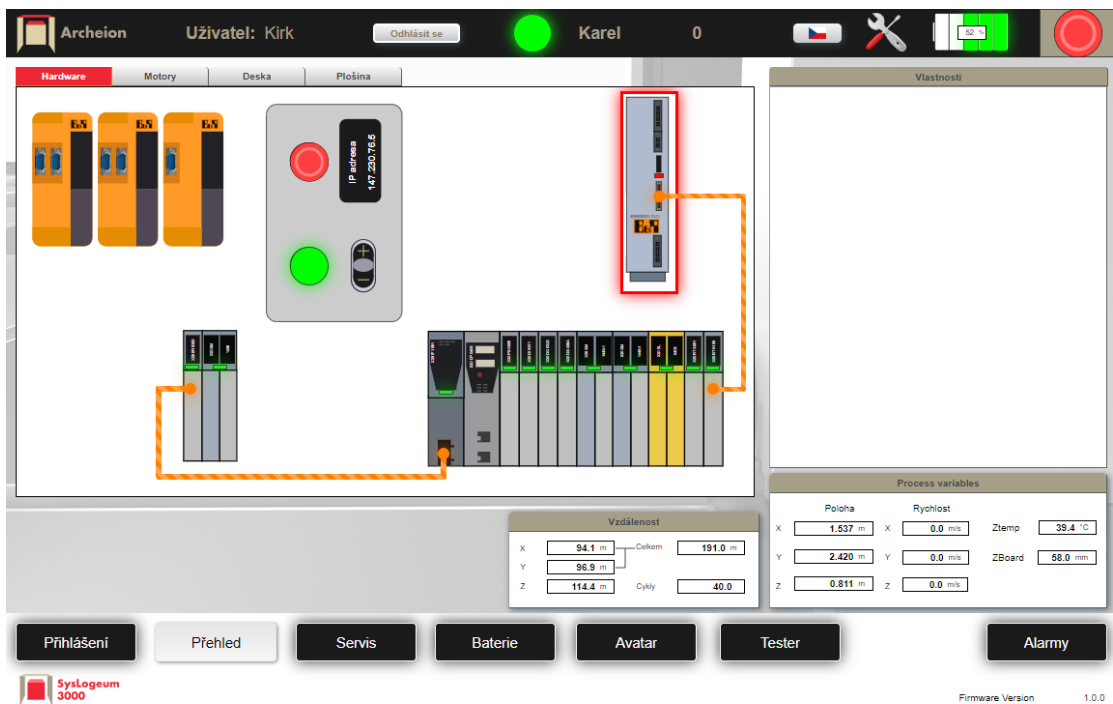
Přihlášení Přehled Servis Baterie Avatar Tester Alarmy

SysLogeum 3000 Firmware Version 1.0.0

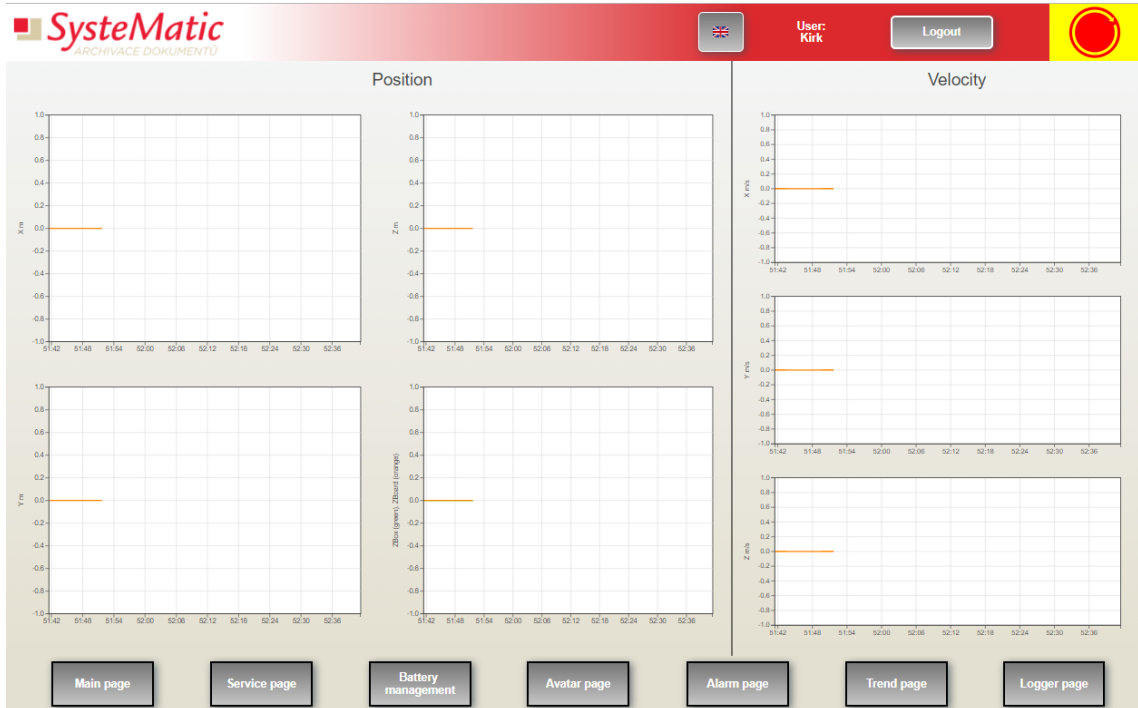
Obrázek A.10: Stránka Alarmy (nová verze)



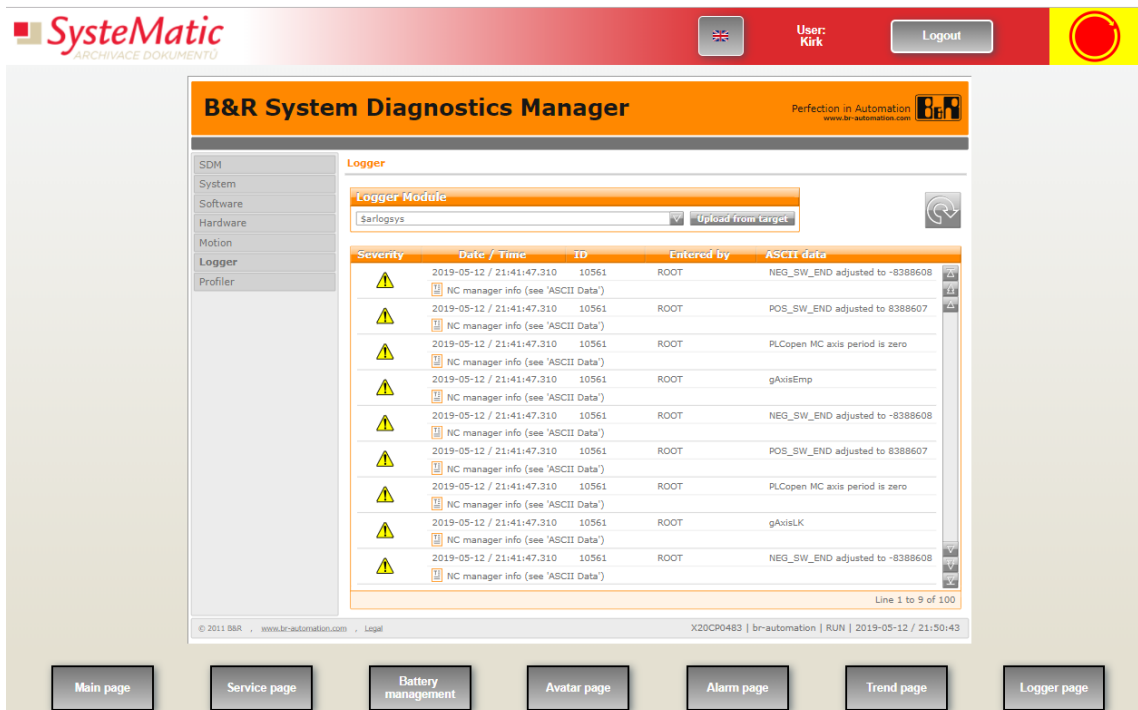
Obrázek A.11: Stránka Tester (přidána)



Obrázek A.12: Stránka Přehled (přidána)



Obrázek A.13: Stránka Trendy (odstraněna)



Obrázek A.14: Stránka Logger (odstraněna)

B Maják

- Svítí zelená - robot je ve stavu READY
- Bliká zelená - robot je ve stavu STAND BY $f = 1$ Hz
- Svítí oranžová - robot je ve stavu WARNING - může nadále pracovat
- Bliká oranžová - probíhá BOOT systému $f = 1$ Hz
- Bliká oranžová - probíhá nabíjení systému $f = 0,5$ Hz
- Svítí červená - robot je ve stavu ERROR - robot nemůže pracovat
- Bliká červená - robot po restartu čeká na vyčtení historie chyb z registru $f = 2$ Hz
- Bliká oranžová- robot po restartu čeká na vyčtení historie varování z registru $f = 2$ Hz

B.1 Tlačítko na Majáku

Potvrzení (smazání) chyb po opětovném zapnutí systému

- výchozí stav: maják bliká červeně s $f = 1$ Hz
- akce: stiskni tlačítko na 4 s
- odezva: po potvrzení začne maják blikat zeleně s $f = 1$ Hz - robot je ve stavu READY

Aktivace servisního režimu

- výchozí stav: maják bliká červeně s $f = 1$ Hz, maják bliká zeleně s $f = 1$ Hz, maják svítí červeně
- akce: stiskni tlačítko na 7 s
- odezva: maják svítí oranžově

Deaktivace servisního režimu

- výchozí stav: maják svítí oranžově

- akce: stiskni tlačítko na 7 s
- odezva: maják bliká zeleně s $f = 1 \text{ Hz}$ - robot je ve stavu READY

Aktivace servisního režimu osy X

- výchozí stav: maják svítí oranžově
- akce: stiskni tlačítko 2x v 5s časovém okně
- odezva: maják bliká zeleně s $f = 1 \text{ Hz}$

Aktivace servisního režimu osy Y

- výchozí stav: maják svítí oranžově
- akce: stiskni tlačítko 3x v 5s časovém okně
- odezva: maják bliká oranžově s $f = 1 \text{ Hz}$

Aktivace servisního režimu osy Z

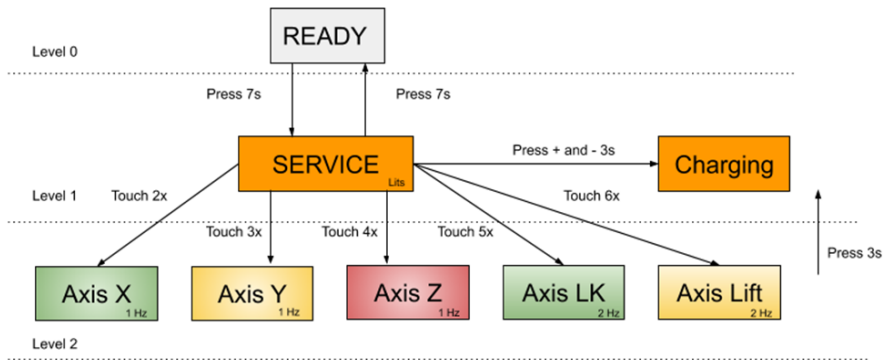
- výchozí stav: maják svítí oranžově
- akce: stiskni tlačítko 4x v 5s časovém okně
- odezva: maják bliká červeně s $f = 1 \text{ Hz}$

Aktivace servisního režimu osy LK

- výchozí stav: maják svítí oranžově
- akce: stiskni tlačítko 5x v 5s časovém okně
- odezva: maják bliká zeleně s $f = 2 \text{ Hz}$

Aktivace servisního režimu osy Lift

- výchozí stav: maják svítí oranžově
- akce: stiskni tlačítko 6x v 5s časovém okně
- odezva: maják bliká oranžově s $f = 2 \text{ Hz}$



Obrázek B.1: Grafické zobrazení volby servisního režimu a signalizace

B.2 Tlačítka a signalizace pro řízení pohybu v servisním režimu

- výchozí stav: maják bliká podle volby osy X viz Obr. B.1
- výchozí stav: LED svítí zeleně - pohon připraven k pohybu LED nesvítí - nelze pohybovat vybranou osou - možný pouze mechanický pohyb
- akce: stisk směr pohybu +
- odezva: zvolená osa se pohybuje v kladném směru

- výchozí stav: maják bliká podle volby osy X viz Obr. B.1
- výchozí stav: LED svítí zeleně - pohon připraven k pohybu LED nesvítí - nelze pohybovat vybranou osou - možný pouze mechanický pohyb
- akce: stisk směr pohybu -
- odezva: zvolená osa se pohybuje v záporném směru

Zapnutí nabíjení

- výchozí stav: maják svítí oranžově - stav SERVICE viz Obr. B.1
- výchozí stav: LED nesvítí
- akce: současný stisk směr pohybu + a směr pohybu - po dobu 3 s
- odezva: baterie se nabíjí odezva: LED bliká $f = 1\text{Hz}$

Vypnutí nabíjení

- výchozí stav: maják svítí oranžově - stav SERVICE viz Obr. B.1

- výchozí stav: LED bliká $f = 1\text{Hz}$
- akce: současný stisk směr pohybu + a směr pohybu - po dobu 3 s
- odezva: baterie se nenabíjí odezva: LED nesvítí

C Návod



Návod k vizualizaci skladovacího robota systému SysLogeum 3000



Obsah

Základní informace	3
Prvky ve vizualizaci	3
Rozložení vizualizace	3
Připojení se k vizualizaci	4
Stránky vizualizace.....	4
Přihlášení	4
Přehled	5
Hardware	6
Motory	7
Deska	7
Plošina	8
Servis	8
Baterie	10
Avatar	10
Tester	11
Alarmy	12
Ovládání robota ze servisního režimu	13

Základní informace

Tento návod obsahuje popis ovládání vizualizace skladovacího robota systému Archeion nebo SysLogeum 3000. Tato vizualizace umožňuje uživateli diagnostikovat robota a ovládat jej pomocí servisního režimu nebo struktury Avatar.

Návod neobsahuje informace potřebné k ovládání robota v rámci skladu. K tomu bude určen návod pro operátora.

Prvky ve vizualizaci

Celá vizualizace obsahuje především tři typy prvků:

- Tlačítka (černé obdélníky, které po kliknutí vykonávají danou funkci a mění barvu)
- Prvky umožňující nastavit hodnotu proměnné (bílé obdélníky s červenými okraji, po kliknutí na ně je zobrazeno okno umožňující nastavit hodnotu)
- Prvky zobrazující hodnotu nebo stav proměnné (bílé obdélníky, barevné čtverečky nebo červené obdélníky s bílým textem)



Obr. 1 Typy prvků

Rozložení vizualizace

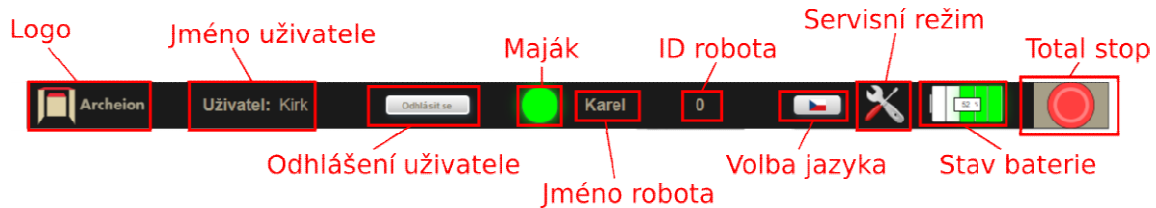
Vizualizace je rozdělena na tři části:

- Horní část (Hlavička)
- Dolní část (Navigace)
- Aktuální zobrazená stránka

Hlavička a Navigace je zobrazena na všech stránkách. Navigace vizualizace umožňuje přepínání jednotlivých stránek vizualizace. Aktuální zobrazená stránka je indikována šedou barvou tlačítka s názvem této stránky.

Obr. 2 Navigace

V Hlavičce je umístěno logo systému Archeion, jméno uživatele, tlačítko pro odhlášení uživatele, vizuální kopie světelného Majáku na robotovi, název robota, ID robota, možnost změny jazyka, dále tlačítko pro spuštění servisního režimu, indikátor stavu baterie a stavu nabíjení baterie a nouzové tlačítko pro zastavení robota Total stop.



Obr. 3 Hlavička

Otevření vizualizace

Pro otevření vizualizace je potřeba spustit libovolný webový prohlížeč (např. Google Chrome) a zadat url adresu vizualizace. K vizualizaci je možné se připojit i prostřednictvím chytrého zařízení.

Url adresa vizualizace se skládá z IP adresy robota, ke kterému se chce uživatel připojit, z portu vizualizace (standartně 81) a z názvu vizualizace (Visu).



Obr. 4 Adresa vizualizace

Stránky vizualizace

Možnosti vizualizace jsou rozděleny na několik stránek. Každá z těchto stránek slouží k jinému účelu.

Přihlášení

Po připojení k vizualizaci je uživateli zobrazena první stránka vizualizace Přihlášení. Na této stránce se nachází prvky pro zadání jména uživatele a hesla. Po zadání těchto údajů je možné provést přihlášení pomocí tlačítka **Přihlásit**.

Uživatel

Heslo

Přihlásit

Obr. 5 Přihlášení

Pokud uživatel zadal správné údaje, zobrazí se zpráva s úspěšným přihlášením a v horní liště vizualizace se zobrazí jméno uživatele. Pokud jsou údaje špatné, zobrazí se zpráva s neúspěšným přihlášením.



Obr. 6 Zprávy pro uživatele

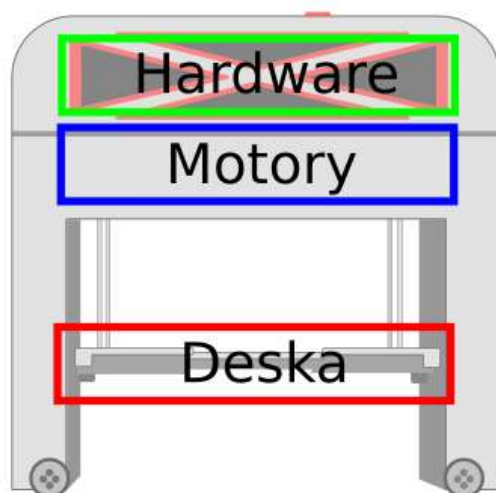
Každý uživatel má definovanou roli. Nepřihlášený uživatel má roli „Everyone“. Tato role umožňuje uživateli zobrazit některé stránky, ale nemá možnost, cokoli měnit nebo ovládat. Po přihlášení může mít uživatel roli „Operator“, „Service“, „Avatar“ nebo „Supervisor“. Každá z těchto rolí dává uživateli jinou pravomoc.

Role	Práva pro				
	Zobrazení stránky Tester	Zobrazení zbylých stránek	Ovládání stránky Servis	Ovládání stránky Avatar	Ovládání stránky Tester
Everyone	Ne	Ne	Ne	Ne	Ne
Operator	Ne	Ano	Ne	Ne	Ne
Service	Ano	Ano	Ano	Ne	Ano
Avatar	Ne	Ano	Ne	Ano	Ne
Supervisor	Ano	Ano	Ano	Ano	Ano

Tab. 1 Práva uživatelů

Přehled

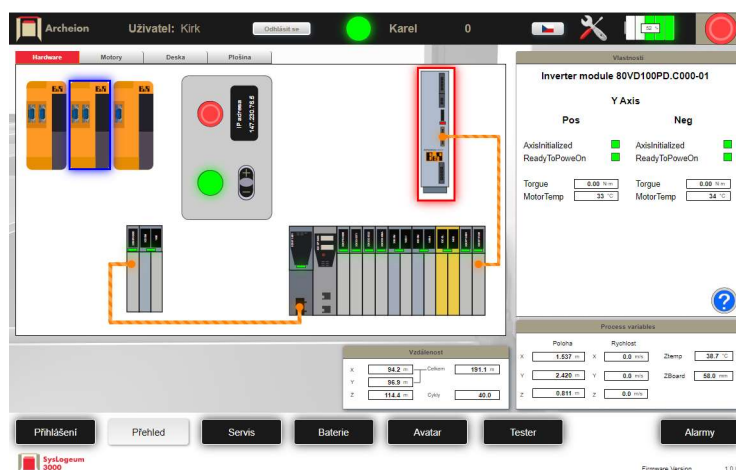
Stránka Přehled zobrazuje některé procesní údaje robota (poloha, rychlost) a jednotlivé hardwarové úrovně robota. Tato stránka umožňuje diagnostikovat možné chyby na jednotlivých modulech robota.



Obr. 7 Hardwarové úrovně robota

Hardware

První úroveň Hardware obsahuje všechny moduly v robotovi (PLC, moduly pro ovládání motorů, modul pro nabíjení atd.). Po kliknutí na jakýkoliv modul je modul označen modře a v okně Vlastnosti jsou zobrazeny proměnné tohoto modulu. Každý modul má navíc možnost

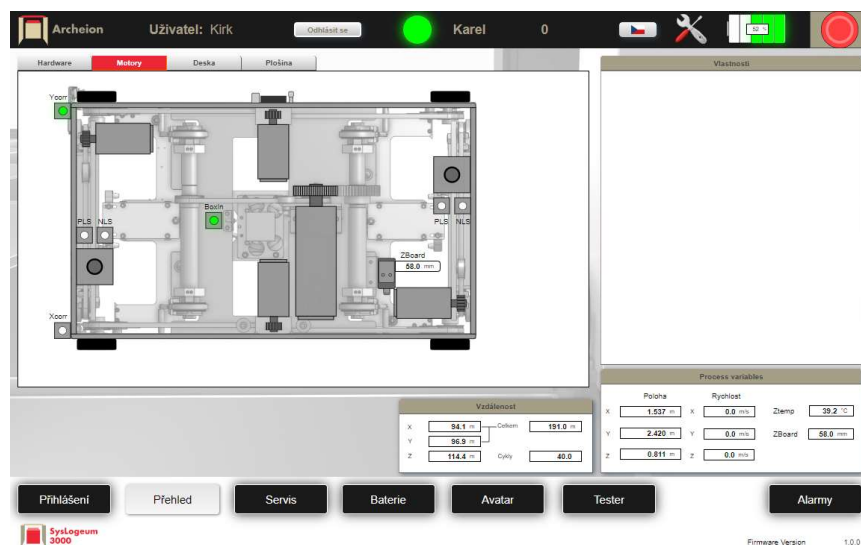


Obr. 8 Hardware

otevření nápovědy. Nápověda obsahuje technické informace o daném modulu.

Motory

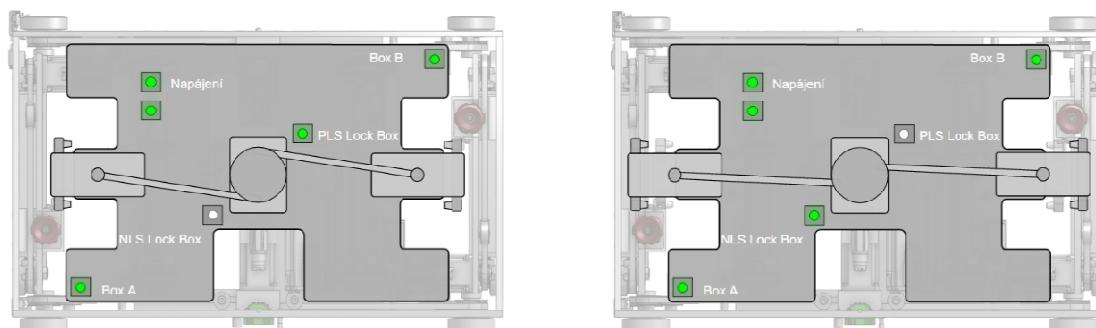
Druhá úroveň Motory zobrazuje rozmístění motorů na robotovi, hodnoty jejich koncových snímačů a senzor vzdálenosti mezi deskou a robotem.



Obr. 9 Motory

Deska

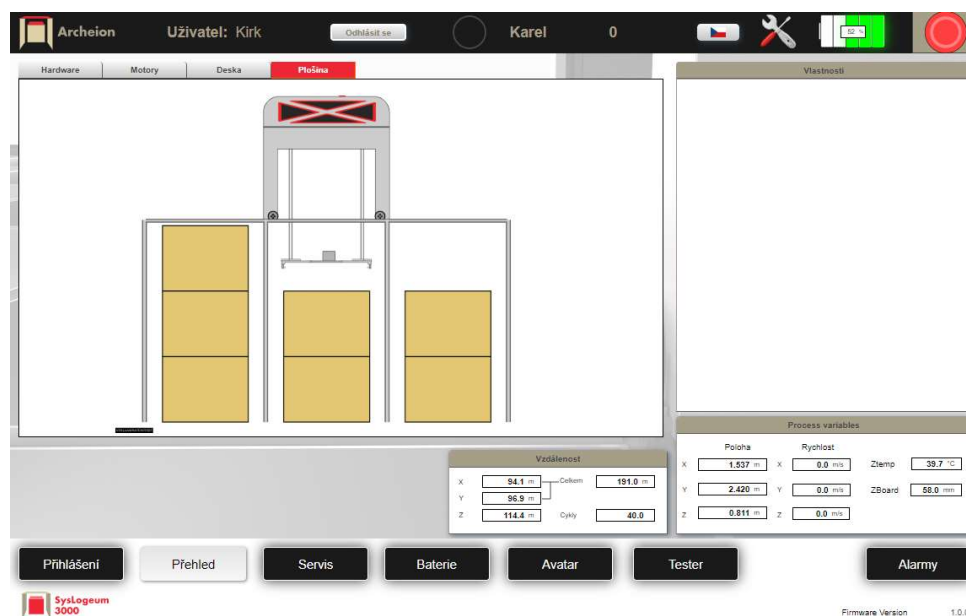
Třetí úroveň Deska obsahuje umístění snímačů na desce robota a krokového motoru, který ovládá uchopovací mechanismus. Aktuální stav uchopovacího mechanismu (uchopení nebo uvolnění) je možné zjistit hodnotou snímačů **PLS Lock box** a **NLS Lock box**. Pokud je mechanismus ve stavu uchopení, svítí zeleně snímač **PLS Lock box**. Navíc se při změně stavu mechanismu změní i zobrazení desky a zobrazuje tak aktuální stav uchopovacího mechanismu.



Obr. 10 Deska

Plošina

Čtvrtá úroveň Plošina ukazuje proces uchopení a položení přepravky. Jednotlivé části animace jsou napojené přímo na skutečné procesní stavy robota. Jedná se tak o vizuální zobrazení stavu robota při pohybu desky v reálném čase.



Obr. 11 Plošina

Servis

Stránka Servis umožňuje uživateli s rolí „Service“ nebo „Supervisor“ ovládat robota v servisním režimu a povolit ovládání ze struktury Avatar. Prvky na stránce jsou umístěny do několika skupin podle jejich účelu:

- **State HW** zobrazuje aktuální hardwarové stavy robota (pohyb robota, stav zdvihu, stav desky atd.).
- **State SW** zobrazuje aktuální softwarové stavy robota (stav nabíjecí stanice nebo stavy programů ovládající jednotlivé části robota)
- **Motion commands** umožňuje pohybovat robotem. Tlačítka **X direction** a **Y direction** přepnou podvozek robota na zvolený směr (X nebo Y). Tlačítko **Z direction** uzamkne podvozek robota na gridu, aby bylo možné bezpečně pohybovat deskou. Po kliknutí na prvky **Velocity** nebo **Position** je možné zadat hodnoty souřadnic, na které se má robot přemístit. Pro jednodušší zadávání souřadnic je možné kliknout na tlačítko **Plus** nebo **Mínus**. Tyto tlačítka nastavují automaticky souřadnice další nebo předchozí skladovací pozice. Po stisknutí **X go** nebo **Y go** se robot začne pohybovat daným směrem na danou souřadnici.

- **Corrections** umožňuje zadat hodnoty korekcí používané robotem pro dosažení přesné požadované pozice.
- **Robot orientation** zobrazuje aktuální orientaci robota vůči souřadnému systému. Tlačítkem **Change orientation** se změní obrázek a i souřadný systém robota. Tlačítko **Set XY position** pozici nastaví robotovi aktuální pozici podle hodnot polohy ve skupině **Motion commands**.
- **Errors and warnings** zobrazuje aktuální aktivní chyby nebo varování. Tlačítka **Error Ack** a **Warning Ack** tyto jednotlivé chyby a varování potvrzují.
- **Binary states** zobrazují hodnoty hardwarových a softwarových binárních snímačů (hodnoty koncových snímačů, připojení nabíječky, orientaci robota).
- **AvatarCOM** povoluje a zakazuje ovládání robota pomocí struktury Avatar.
- **Distance** ukazuje vzdálenost, kterou robot ujel v jednotlivých osách a počet cyklů uchopení a položení přepravky.
- **Charging** umožňuje uživateli spustit nebo zastavit nabíjení robota z nabíječky a zobrazuje hodnoty baterie.
- **Process variables** vypisuje aktuální hodnoty pozice a rychlosti robota v jednotlivých osách a teplotu PLC robota.
- **Z axis** zpřístupňuje uživateli ovládání desky a uchopovacího mechanismu. Údaj **Velocity fast** určuje rychlost pohybu desky, **Open level** vzdálenost, kdy se má otevřít uchopovací mechanismus, od přepravky, **Park level** vzdálenost od přepravky, kdy má deska zpomalit, a údaj **Z lay down** je výška umístění přepravky.

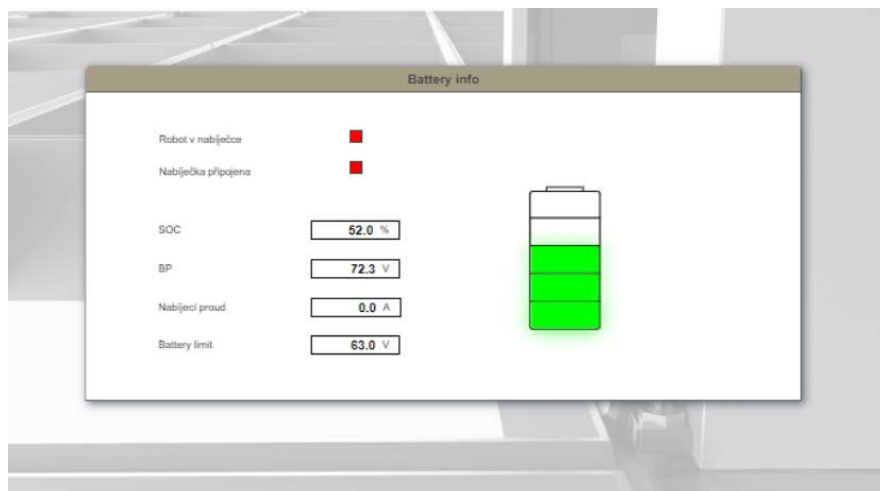
V horní části této stránky je pět tlačítek. Tlačítko **POWER OFF** vzdáleně vypíná napájení robota, tlačítko **Reset PLC** resetuje PLC, tlačítka **ERR RESET** A **WAR RESET** potvrzují všechny chyby a varování naráz.



Obr. 12 Stránka Servis

Baterie

Stránka Baterie slouží pouze k zobrazování údajů baterie.



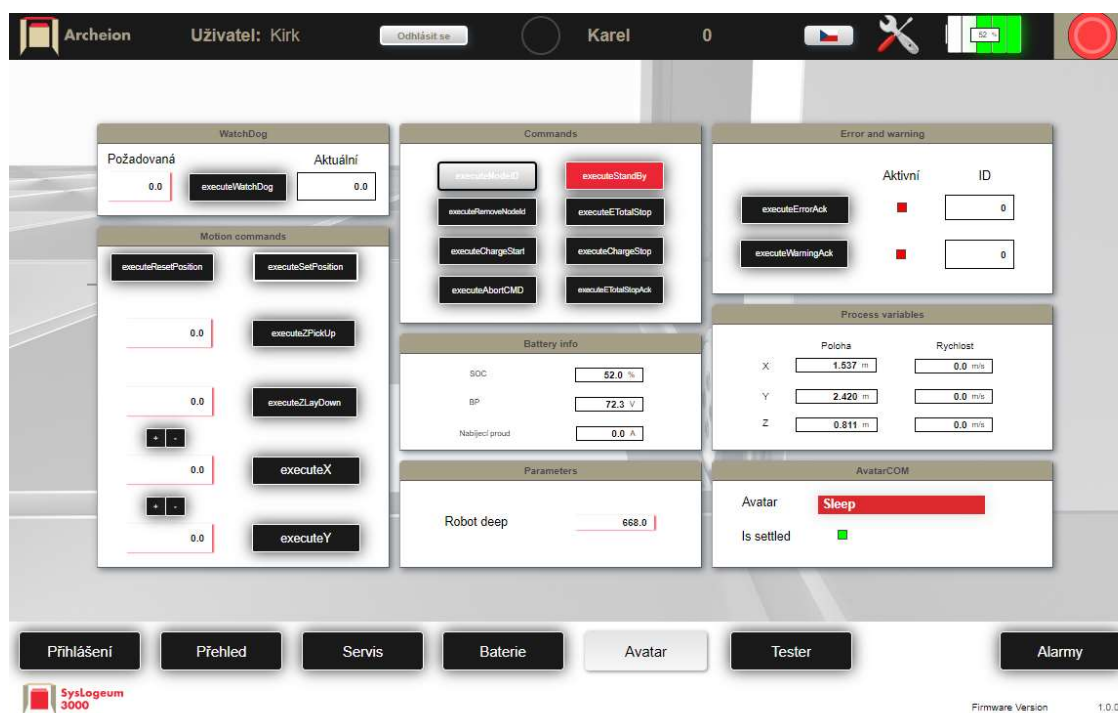
Obr. 13 Stránka Baterie

Avatar

Stránka Avatar slouží k ovládní robotu ze struktury Avatar. Tato stránka simuluje příkazy, které dostává robot od nadřazeného systému. Použitím této stránky je možné zjistit, zda komunikace mezi robotem a nadřazeným systémem funguje správně. Oproti stránce Servis jsou zde jenom základní příkazy pro ovládní robotu. Jenom uživatel s rolí „Avatar“ nebo „Supervisor“ může tuto stránku použít. Stejně jako na stránce Servis jsou i zde prvky umístěny do několika skupin:

- **WatchDog** slouží ke kontrole komunikace přes OPC UA server mezi robotem a nadřazeným systémem. Uživatel změní hodnotu prvku **Required** a stiskne tlačítko **executeWatchDog**. Pokud se změní hodnota prvku **Aktual** na hodnotu prvku **Required**, tak mezi robotem a nadřazeným systémem probíhá komunikace.
- **Commands** obsahuje příkazy pro robota jako je např. spuštění nabíjení, zastavení nabíjení, přepnutí robota do režimu Stand by aj.
- **Error and warning** je stejný jako **Errors and warnings** na stránce Servis. Jsou zde potvrzovány a zobrazovány chyby a varování.
- **Motion commands** umožňuje zadat požadovanou hodnotu pozice v ose X nebo ose Y a pohnout robotem na tuto pozici (**executeX** , **executeY**), nastavit výšku zvednutí (**executeZPickUp**) nebo položení přepravky (**executeZLayDown**) a nastavit aktuální pozici robota (**executeSetPosition**, **executeResetPosition**).
- **Battery info** zobrazuje informace o baterii.
- **Process variables** vypisuje aktuální polohu a rychlost robota v jednotlivých osách.

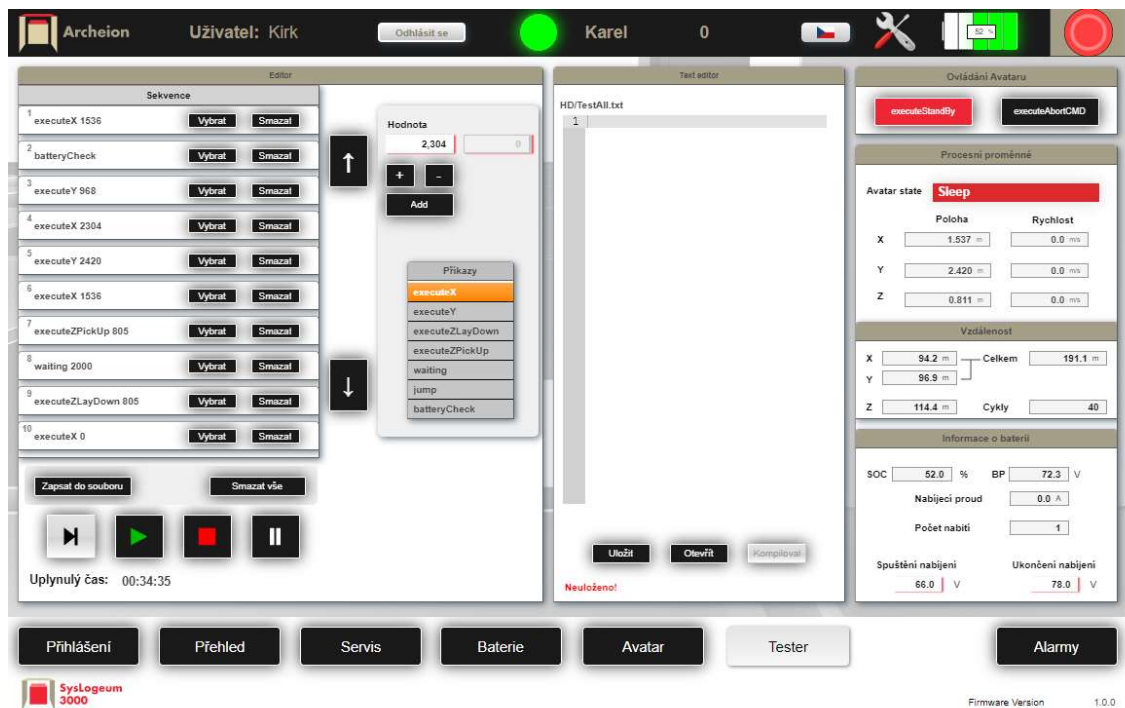
- **Parameters** určuje vzdálenost desky a nulové souřadnice pro ukládání přepravek (**Robot deep**)
- **AvatarCOM** popisuje stav programu Avatar.



Obr. 14 Stránka Avatar

Tester

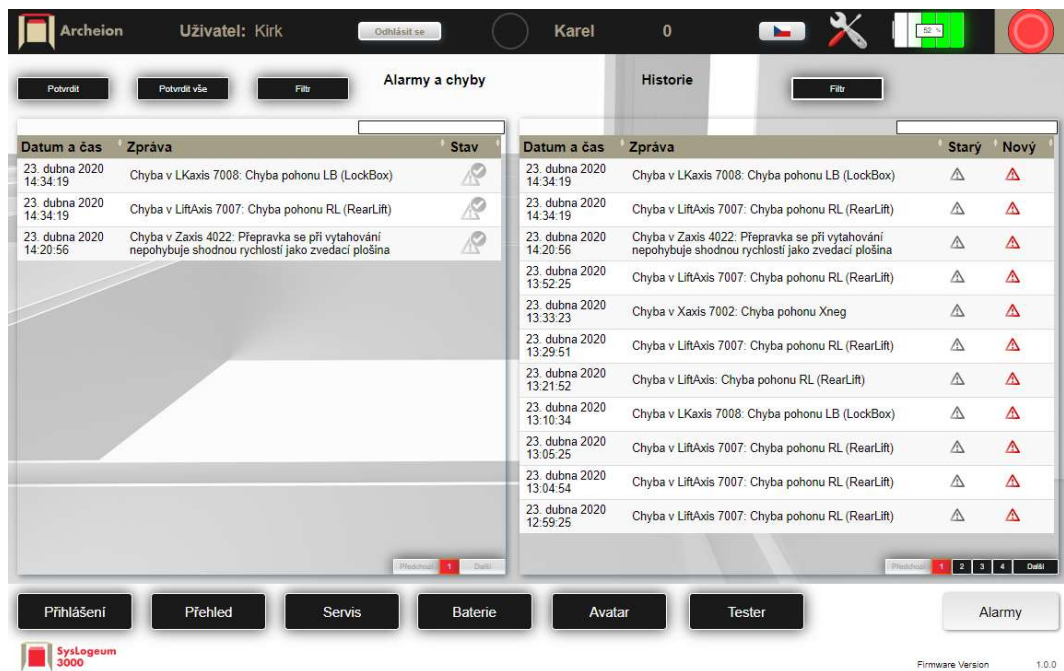
Tester umožňuje uživateli s rolí „Service“ nebo „Supervisor“ zadat robotovi jednotlivé úkony, které pak bude robot vykonávat cyklicky. Pro přesnější popis této stránky a způsob ovládání viz manuál Servisní tester.



Obr. 15 Stránka Tester

Alarmy

Poslední stránka Alarmy zobrazuje všechny aktivní a nepotvrzené chyby a varování a všechny změny stavů jednotlivých chyb a varování. Pomocí tlačítek **Potvrdit** a **Potvrdit vše** je možné chyby a varování potvrdovat buď jednotlivě, nebo všechny naráz. Tlačítko **Filtr** otevře okno, ve kterém je možné nastavit filtry zobrazených chyb a varování.



Obr. 16 Stránka Alarmy

Ovládání robota ze servisního režimu

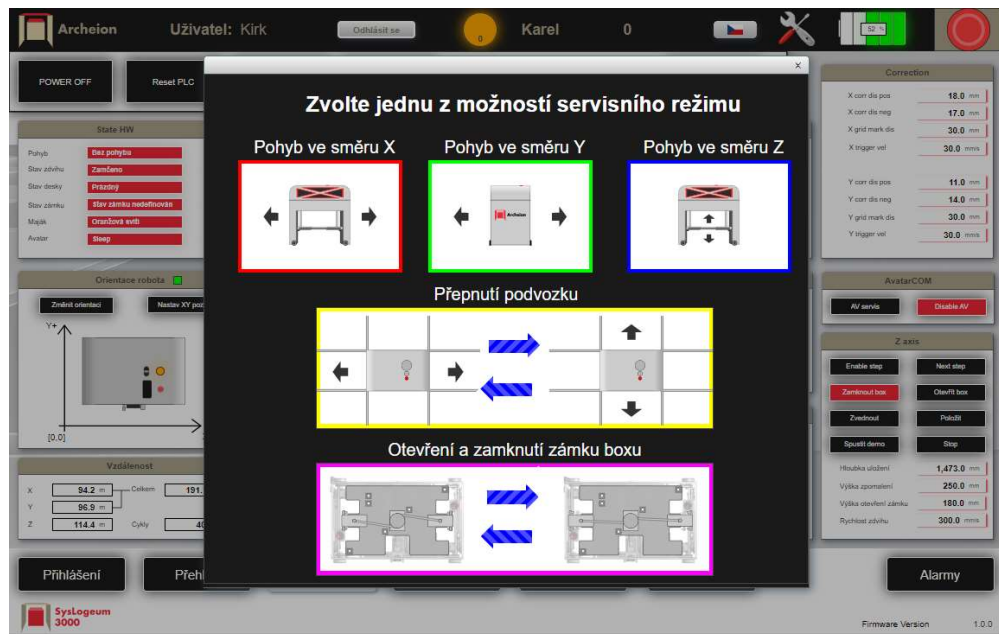
Uživatelům s rolí „Service“ nebo „Supervisor“ je po přihlášení v pravé části Hlavičky vizualizace zpřístupněn prvek pro spuštění servisního režimu. Pomocí tohoto prvku je možné ovládat jednotlivé osy robota v situaci, kdy nastane provozní chyba robota, kterou nelze vyřešit standardním ovládáním.



Obr. 17 Prvek pro spuštění servisního režimu

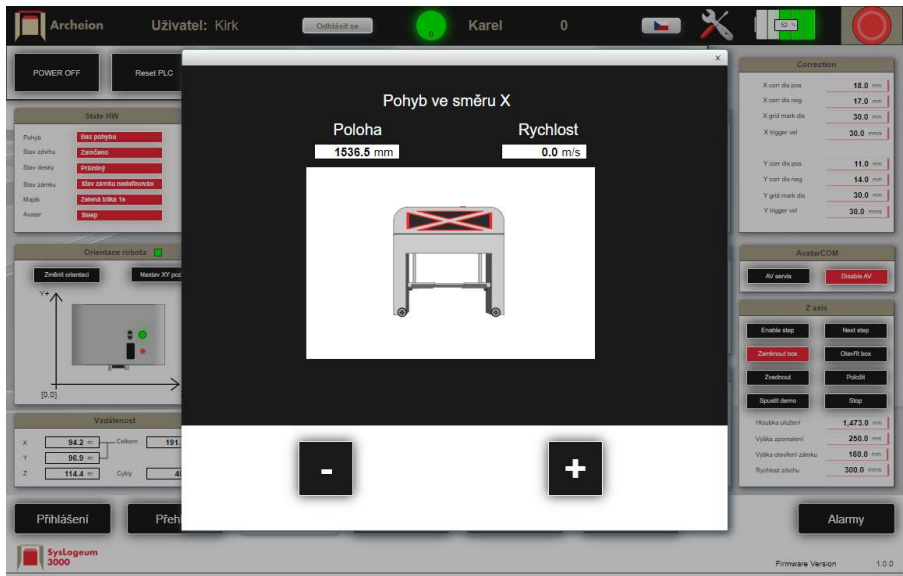
Po kliknutí na tento prvek je robot přepnut do servisního režimu (Maják svítí oranžově) a ve vizualizaci se otevře nové okno. V tomto okně je možné zvolit požadovanou osu robota k ovládání. Je možné zde zvolit:

- Ovládání motorů na ose X (**X direction movement**)
- Ovládání motorů na ose Y (**Y direction movement**)
- Pohybování deskou nahoru a dolů (**Z direction movement**)
- Přepnutí podvozku ze směru X na směr Y nebo naopak (**Lift change**)
- Otevření nebo uzavření uchopovacího mechanismu (**Open a lock board**)



Obr. 18 Možnosti servisního režimu

Po kliknutí na danou možnosť servisného režimu sa robot prepne do servisného ovládání zvolené osy. Zobrazí sa nové okno určené k ovládání této osy. Pro každou osu se zobrazí jiné okno. Ovládání dané osy je možné použitím tlačítka **Plus** nebo **Mínus**.



Obr. 19 Pohyb ve směru X