



Bakalářská práce

Webové rozhraní na PLC Siemens S7-1200 pro dobíjecí stanici

Studijní program:

B0613A140005 Informační technologie

Studijní obor:

Aplikovaná informatika

Autor práce:

Bohumil Hruška

Vedoucí práce:

Ing. Lukáš Hubka, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



Zadání bakalářské práce

Webové rozhraní na PLC Siemens S7-1200 pro dobíjecí stanici

<i>Jméno a příjmení:</i>	Bohumil Hruška
<i>Osobní číslo:</i>	M20000059
<i>Studijní program:</i>	B0613A140005 Informační technologie
<i>Specializace:</i>	Aplikovaná informatika
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Seznamte se s možnostmi webového serveru implementovaného v PLC S7-1200.
2. Prozkoumejte možnosti tvorby dynamických komponent pro web server PLC S7-1200.
3. Vytvořte aplikaci pro webové HMI dobíjecí stanice pro potřeby koncového uživatele i správce a otestujte ji (na PC, v mobilu).
4. Vytvořte knihovnu komponent a funkcí pro TIA STEP7 pro potřeby webového HMI dobíjecí stanice.
5. Sestavte podrobnou dokumentaci a metodický pokyn pro vytvoření vlastních stránek.

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

Seznam odborné literatury:

- [1] Creating User-Defined Web Pages for S7-1200/S7-1500 [online]. V4.0 03/2020. Siemens, 2020 [cit. 2022-10-17]. Dostupné z: https://support.industry.siemens.com/cs/attachments/68011496/68011496_S7-1200_1500_Webserver_DOC_v4_en.pdf
- [2] Basics on Creating HTMLs for SIMATIC CPUs: Application Description [online]. V1.0 02/2014. Siemens, 2014 [cit. 2022-10-17]. Dostupné z: https://support.industry.siemens.com/cs/attachments/68011496/68011496_html_basics_for_simatic_cpus_en.pdf
- [3] Creating and using user-defined web pages on S7-1200 / S7-1500: ID: 68011496. Industry Support Siemens [online]. Siemens, 2020, 04/07/2020 [cit. 2022-10-17]. Dostupné z: <https://support.industry.siemens.com/cs/document/68011496/creating-and-using-user-defined-web-pages-on-s7-1200-s7-1500?dti=0&lc=en-SK>

Vedoucí práce: Ing. Lukáš Hubka, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černohorský, Ph.D.
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.

Poděkování

Tímto bych rád poděkoval mému vedoucímu práce Ing. Lukášovi Hubkovi, Ph.D. za odbornou pomoc, cenné rady při vývoji, důkladnou kontrolu při organizaci mého projektu a hodnotné zkušenosti do budoucího studia.

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací ovládacího webového rozhraní pro dobíjecí stanici automobilů prostřednictvím PLC Siemens S7-1200. Cílem práce je umožnit uživateli pohodlné řízení dobíjecí stanice z jakéhokoliv zařízení s webovým prohlížečem.

První část práce se zaměřuje na analýzu požadavků na dobíjecí stanici a jejího ovládání pomocí PLC. Dále je provedena rešerše již existujících návrhů webových rozhraní pro PLC Siemens S7-1200 a popsána problematika při vývoji webových aplikací.

V druhé části práce je navrženo webové rozhraní pro dobíjecí stanici, které je navrženo tak, aby umožnilo uživateli pohodlné ovládání dobíjecí stanice, včetně nastavení požadovaného výkonu či konečné kapacity baterie, čtení měřených hodnot napětí, výkonu proudu či zobrazení stavu nabití.

V poslední části práce je popsán metodický návod pro tvorbu a implementaci webového rozhraní pro PLC Siemens S7-1200. Metodická příručka obsahuje práci s TIA portál, tvorbu proměnných a následné propojení webového rozhraní a PLC.

Klíčová slova: PLC, dobíjecí stanice, webový server, programování, webová aplikace

Abstract

This bachelor thesis deals with the design and implementation of a control web interface for a car charging station using PLC Siemens S7-1200. The aim of the thesis is to enable the user to conveniently control the charging station from any device with a web browser.

The first part of the work focuses on the analysis of the requirements for the charging station and its control by PLC. Furthermore, a research of existing web interface designs for Siemens S7-1200 PLC is performed and the issues in the development of web applications are described.

In the second part of the thesis, a web interface for the charging station is designed, which is designed to enable the user to conveniently control the charging station, including setting the desired power or battery capacity, reading measured values of voltage, current, and displaying the state of charge.

The final part of the thesis describes a methodical guide for creating and implementing a web interface for PLC Siemens S7-1200. The methodical guide includes working with the TIA portal, creating variables, and subsequently linking the web interface and PLC.

Keywords: PLC, charging station, web server, programming, web application

Obsah

1	Úvod	12
2	Studie případů.....	14
2.1	Možnosti webového rozhraní pro PLC	14
2.2	Bezpečnost webových aplikací	16
2.3	Problematika zobrazení dat uživateli	17
2.4	Škálovatelnost grafického rozhraní	18
2.5	Náročnost webové aplikace.....	19
3	Hardware	20
3.1	PLC obecně	20
3.2	PLC Siemens S7-1200	20
4	Software a použité technologie	22
4.1	TIA Portal.....	22
4.2	Ajax	22
4.3	JQuery	22
4.4	Výměna dat mezi klientem a serverem	22
4.5	Možnosti zabezpečení webové aplikace	24
5	Vývoj a testování	26
5.1	Návrh aplikace	26
5.2	Řídící logika programu	26
5.3	Výměna dat	27
5.4	Grafické rozhraní	27
5.5	Testování webové aplikace	30
6	Tvorba knihovny komponent	31
6.1	Tvorba datového bloku	31
6.2	Napojení datových bloků	33
7	Metodika tvorby vlastních stránek	35

7.1	Inicializace HTML dokumentu	36
7.2	Vytvoření datových bloků.....	37
7.3	Vytvoření proměnných.....	37
7.4	Čtení a zápis proměnných	38
7.5	Čtení a zápis z DB www stránek.....	38
7.6	Čtení a zápis z libovolného DB.....	39
7.7	Zobrazení dat.....	40
7.8	Statické zobrazení dat	41
7.9	Dynamické zobrazení dat	43
8	Závěr.....	44
9	Seznam použité literatury	45

Seznam zkratek

HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
LAN	Local Area Network
WAN	Wide Area Network
DB	Datový blok
SPA	Single Page Application
AJAX	Asynchronous JavaScript And XML

Seznam obrázků

Obrázek 1 Schéma Internet of Things.....	12
Obrázek 2 Popis vývoje webové aplikace.....	14
Obrázek 3 PLC Siemens S7-1200.....	20
Obrázek 4 Schéma HTTP protokolu.....	23
Obrázek 5 Schéma Firewall brány	25
Obrázek 6 GUI – mobilní verze	28
Obrázek 7 GUI – desktopová verze	29
Obrázek 8 Nejpoužívanější funkce pro konverzi datových typů	33
Obrázek 9 Příklad použití funkce CONV	34
Obrázek 10 Příklad použití funkce MOVE.....	34

1 Úvod

V posledních letech technologického rozvoje se většina lidí a firem soustředí na práci skrze internet věcí. Slovo pochází z anglického slova Internet of Things (IoT) a jedná se o vše spojené s komunikací pomocí internetu. Během vývoje došlo k výrazným změnám, a proto mobily, které nám připadaly kdysi jako obrovský zázrak, jsou dnes pro nás již jako nezbytná součást života, která je přirozená. K internetu se dnes nemusíte připojit pouze pomocí mobilu či počítače, ale také pomocí trouby, myčky, pračky nebo pracovních nástrojů. Většina lidí už jistě zná pojmy jako SmartHome, SmartTV nebo SmartHealth a hojná většina z nich je používá a ani si toho nejsou vědomy. V dnešní době není problém si z pohodlí domova nastartovat auto či z práce otevřít psovi dveře na dvorek. V tomto ohledu nabírá technologie na užitečnosti nejen v domácnosti, ale i ve zdravotnictví, zemědělství, architektuře, dopravě či průmyslu.



Obrázek 1 Schéma Internet of Things

Využití elektrických vozidel se v dnešní době stává stále populárnějším, a to nejen díky ekologické stránce.

Avšak použití těchto automobilů závisí na dostupnosti a přístupnosti dobíjecích stanic. Jako součást modernizace a rozvoje infrastruktury nabíjení, je kladen důraz na stavbu dobíjecích stanic, které jsou snadno přístupné a ovladatelné pro uživatele. Jedná se o velký pokrok v technologii, a proto se tato práce dá v budoucnu použít souběžně s chytrou domácností. [13]

Tato bakalářská práce se věnuje návrhu a implementaci ovládacího webového rozhraní pro dobíjecí stanici automobilů prostřednictvím PLC Siemens S7-1200. Cílem této práce je vytvořit uživatelsky přívětivé rozhraní pro obsluhu nabíjecí stanice, které bude dostupné z libovolného zařízení s webovým prohlížečem.

První část práce je zaměřena na analýzu požadavků a přehled již existujících návrhů webových rozhraní pro daný PLC systém. V další části je popsán samotný návrh a implementace webového rozhraní pro dobíjecí stanici, které umožňuje uživateli snadné ovládání a řízení nabíjecí stanice pomocí webové aplikace. V následující části je zahrnuta tvorba knihovny komponent pro TIA Portal, která popisuje tvorbu programových a datových bloků pro budoucí napojení na webové aplikace. V poslední části je popsán metodický postup pro tvorbu a implementaci webového rozhraní pro PLC Siemens systémy včetně popsání základní práce s TIA portálem a následným propojením s webovým rozhraním. Celkovým výsledkem této práce je funkční webová aplikace pro snadné ovládání dobíjecí stanice pomocí libovolného zařízení s webovým prohlížečem.

2 Studie případů

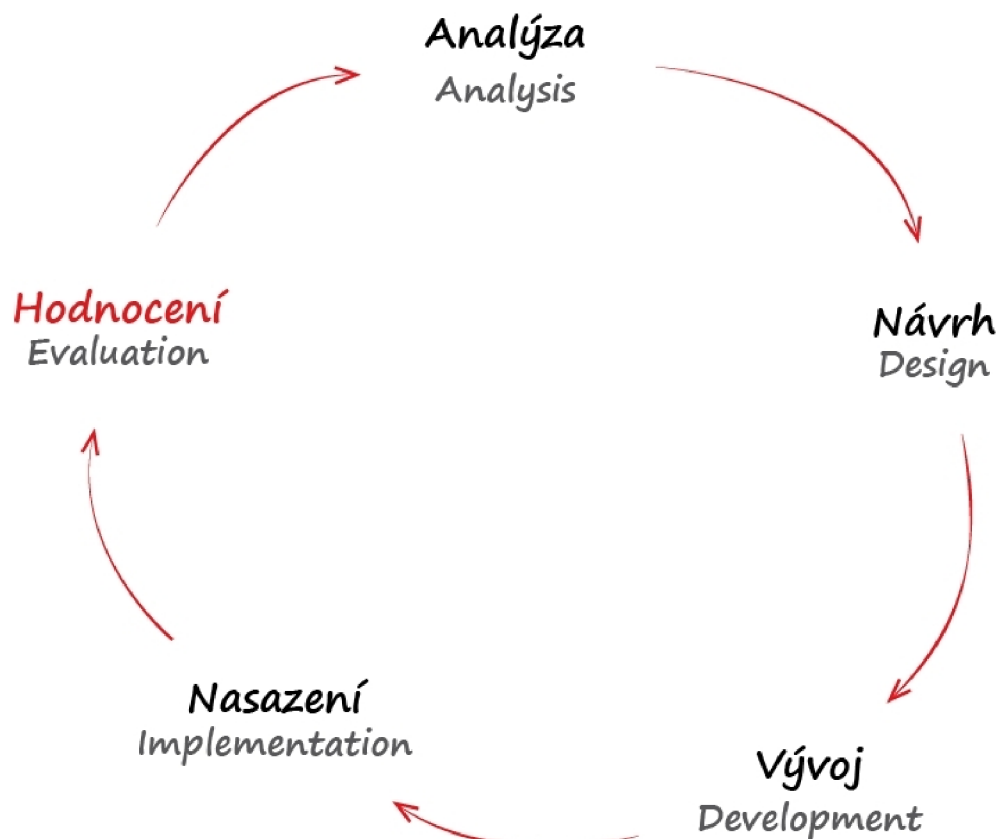
2.1 Možnosti webového rozhraní pro PLC

Webové rozhraní pro PLC (Programmable Logic Controller) poskytuje uživateli vzdálené ovládání a přístup k řídicímu systému za pomoci webového prohlížeče. Je mnoho způsobů, jak implementovat webové rozhraní pro PLC. Samotné webové rozhraní na PLC může uživateli nabízet:

Vzdálený monitoring a řízení – sledování stavů zařízení a ovládání z jakéhokoli místa s připojením k internetu.

Diagnostika a opravy – zobrazit uživateli chybové stavy a identifikovat problémy a tím zajistit lepší produktivitu a menší chybovost.

Nastavení a aktualizace – umožňuje uživateli efektivně konfigurovat a aktualizovat systém bez nutnosti fyzické přítomnosti u zařízení.



Obrázek 2 Popis vývoje webové aplikace

Mezi dostupné možnosti implementace webového rozhraní patří:

1. Vestavěný webový server – jedná se o integrovaný webový server v některých moderních PLC, umožňuje přístup k webovému rozhraní přímo z PLC, nevyžaduje další externí zařízení.
2. Externí webový server – v případě kdy PLC neobsahuje webový server, je možné připojit webový server k PLC a umožnit tak přístup k řídicímu systému. Je zde však nutná potřeba dalšího hardware a software, ale může nabídnout rozsáhlejší funkce a flexibilitu.
3. Speciální webové rozhraní – jedná se o speciální webový server navržen přímo pro speciální aplikace a nabízí speciální funkce.
4. RESTful API – jedná se o soubor aplikací a služeb přes které je možné ovládat řídicí systém PLC, komunikace probíhá pomocí RESTful API (Representational State Transfer).

Všechny tyto možnosti umožňují vzdálené řízení a monitorování PLC pomocí webového prohlížeče. Použití technologie vždy závisí na konkrétních požadavcích a funkčnosti systému PLC.

Mezi hlavní výhody implementace patří výše zmíněné vzdálené řízení, které přináší uživateli jednoduchost ovládání, změny stavů, zobrazení chybových stavů, a to bez nutnosti fyzické přítomnosti u spravovaného zařízení. Jedna z dalších výhod je nezávislost na použitém zařízení (flexibilita), vzdálená správa je možná jak z mobilního telefonu, tabletu, tak i stolního počítače či notebooku. Mezi nevýhody patří závislost na internetovém připojení – v případě výpadku či ztracení internetového připojení je nemožnost ovládání, což má negativní dopad na produkci. V dnešní době je toto riziko stále menší (veřejné bezdrátové připojení, zvyšování pokrytí mobilní sítě operátora). Ovládání skrze internet přináší i bezpečnostní rizika, jako zvýšené riziko možnosti webového útoku. Je tak nutné dbát na celkovou bezpečnost jak webové aplikace, tak samotného PLC (citlivá data, přístupové údaje, připojení k síti). Pokud celková náročnost webové aplikace překročí mez, kdy PLC není schopné poskytnout dostatečný výpočetní výkon, je zde nutnost dalšího hardware a software.

2.2 Bezpečnost webových aplikací

Webové aplikace patří v dnešní době mezi nezbytnou součást každodenního života a podnikání. Jelikož mnoho z nich obsahuje citlivá data uživatelů, dochází častěji k útokům na tyto aplikace. Útočníci se snaží získat citlivá data, a ty následně použít k zneužití či vydírání dotčených uživatelů, ale i organizací. Proto mezi nezbytnou součástí webových aplikací patří bezpečnostní opatření. [4]

Mezi hlavní mechanismy obrany proti útokům a zneužití citlivých dat patří:

1. Autentizace a autorizace – prvním krokem zabezpečení webové aplikace je zajistit, aby uživatelé, kteří aplikaci používají, byli řádně ověřeni. Tento postup se nazývá „Autentizace“. Autentizace ručí za ověření i ověření přístupu k aplikaci. Dále je pak třeba provádět autorizaci – ověření, zda uživatel má přístup pouze ke konkrétní části aplikace.
2. Šifrování dat – jedná se o přenos šifrovaných dat mezi klientem a serverem. Může být dosaženo použitím HTTPS protokolu či SSL/TLS protokolu.
3. Aktualizace – je důležité, aby aplikace a její součásti byly stále aktualizované. Instalováním aktualizací dosáhneme preventivní ochrany, jelikož i samotní výrobci technologii hledají chyby, které pak opravují.
4. Logování – je možné využít pro sledování co se děje na pozadí běhu aplikace. Výhodou je zpětná kontrola a sledování co se kde stalo.
5. Zálohování – aplikace by měla být pravidelně zálohována. V případě havárie či výpadku systému lze data obnovit z poslední zálohy před havárií.

I přes všechny zmíněné mechanismy ochrany dochází stále k útokům a zneužívání citlivých dat. V dnešní době je téměř nemožné zajistit úplné zabezpečení, aby nemohlo dojít k žádnému útoku. Proto je dobré zajistit prevenci a v případě útoku jej náhle odhalit a případně odstavit.

2.3 Problematika zobrazení dat uživateli

Při zobrazování dat uživateli je více způsobů, jak tyto data zobrazit. Mnoho firem využívá zobrazovací panely (HMI, Monitor, Televize), aby poskytly svým zaměstnancům grafický přehled o chodu činnosti. Velkou výhodou tohoto zobrazení je kompletnost a přehlednost dat na jednom místě, avšak v některých případech to nemusí být žádané. Problém nastává, pokud chceme ovládat či zobrazovat data vzdáleně – při dobíjecí stanici obzvlášť, jelikož chodit se přesvědčit k samotné nabíjecí stanici, zda se auto opravdu nabíjí, nebo pokud budeme chtít nabíjení vypnout, se jeví pro uživatele zbytečné, nepohodlné a neefektivní.

Mezi nevýhody zobrazovacích panelů tedy patří nejen nemožnost vzdáleného ovládání, ale například licence na zobrazení, které jsou často velmi drahé. Tyto licence se mohou vázat na zobrazovací zařízení, ale i na počet zobrazovaných hodnot na zobrazovacím zařízení. V tomto by mohl konečný uživatel vidět velký problém z důvodů vysoké počáteční investice, a proto se mnoho uživatelů soustředí na webové rozhraní.

Další problém nastává při návrhu a implementaci vlastního webového rozhraní. I když mnoho výrobců PLC podporuje tvorbu vlastních stránek a následné nahrání na interní webový server, tak samotný návod výrobce popisuje nanejvýš zobrazení dat do html stránky. Tímto se stává zákazník nevědomě lapen do reklamní sítě a je nucen přikoupit další licence pro zobrazení více dat. Mezi nevýhody interního serveru patří malá výkonnost, a proto je potřeba, co nejvíce zátěže přesunout na klientskou část.

2.4 Škálovatelnost grafického rozhraní

S postupem času a pokročilým vývojem technologií dochází k častějšímu používání webových aplikací na mobilních zařízeních, a proto je třeba tyto aplikace optimalizovat pro zobrazení na různých mobilních zařízeních. Pro optimalizaci se využívá mnoho způsobů tak, aby webové aplikace fungovaly efektivně jak na počítači, tak i na tabletu či chytrém telefonu.

Mezi jednu z nejpoužívanějších možností patří „Responsivní design“. Tento způsob přístupu k optimalizaci grafické vizualizace spočívá v přizpůsobení obsahu webové aplikace podle velikosti obrazovky zařízení. Všechny prvky stránky (tlačítka, obrázky, ovládací prvky, ...) se přizpůsobí tak, aby byly snadno čitelné a ovladatelné na různých zařízeních. [5]

Další možností, jak zajistit responsivitu webové aplikace, je vytvoření tzv. „Mobilní verze“. Mobilní verze aplikace se obvykle liší od desktopové (počítačové verze) tím, že má jednodušší a přehlednější rozhraní. Toto rozhraní je pak snadno ovladatelné na menších obrazovkách. Často se u mobilních verzí využívá zobrazení pouze nejpodstatnějších prvků obsahu, aby byla minimalizována potřeba skrolování (posun obrazovky nahoru a dolů).

Po vytvoření grafického rozhraní je zapotřebí webovou aplikaci otestovat na různých zařízeních. Pro dosažení maximálního výsledku je vhodné aplikaci otestovat na co největším počtu zařízení. To znamená, že aplikace by měla být otestována na různých velikostech obrazovek a operačních systémech (iOS, Android, Windows, ...). Testování by mělo zahrnovat i různé prohlížeče [5] (každý prohlížeč podporuje jiné technologie, jinak vykresluje prvky stránky). Těmito kroky se zajistí, že webová aplikace bude fungovat na všech platformách.

2.5 Náročnost webové aplikace

Náročnost webové aplikace na server se liší v závislosti na mnoha faktorech. Mezi nejvýznamnější patří například počet uživatelů, kteří se k webové aplikaci připojují, počet požadavků na server nebo využití paměti a procesoru samotného serveru. Pro určení náročnosti webové aplikace se používá mnoho metod. Mezi sledované parametry patří čas odezvy, počet požadavků/dotazů za sekundu, které je schopen server obsloužit za sekundu či využití paměti a procesoru paměti. Tyto parametry jsou ukazatelem, jak efektivně zvládá server zpracování zátěže.

Při tvorbě samotné aplikace je potřeba brát zřetel na optimalizaci odesílaných požadavků na server a uživatelskou přívětivost. V případě, kdy se jedná o webové aplikace pro PLC servery, je náročnost převáděna na stranu klienta. Hlavním faktorem pro přenos náročnosti na stranu klienta je, že samotné interní PLC servery nejsou dosti výkonné, aby na nich běžely interaktivní aplikace či probíhal náročný výpočet. Samotné PLC jsou zaměřené na výpočet a řízení celistvého systému, a proto jejich webový server slouží převážně pro převod dat klientovi.

Při tvorbě elementární webové aplikace na zobrazení hodnoty z interní paměti PLC je tato metoda dostačující. Je zde minimální požadavek na náročnost na server – klient pošle požadavek, server zpracuje požadavek, pošle data zpět na klienta a klient je dále zpracuje pro zobrazení uživateli. Pokud bychom však chtěli aplikaci se složitou a rozsáhlou databází nebo velkým počtem připojovaných uživatelů, tak v tomto případě už nám interní server nestačí.

3 Hardware

3.1 PLC obecně

Jedná se o víceúčelový modul neboli programovatelný logický automat (dále jen PLC). Jedná se o zařízení, která jsou hojně využívána v průmyslovém prostředí pro řízení v reálném čase. Mezi přednosti zařízení patří vysoká spolehlivost, odolnost vůči vnějším vlivům, jednoduchá diagnostika, možnost editace řídicích algoritmů či rychlá realizace. Samotná komunikace mezi PLC a řízeným objektem je realizovaná pomocí vstupních a výstupních komunikačních modulů.

3.2 PLC Siemens S7-1200

Jako základní stavební modul celého projektu je použito PLC Siemens S7-1200 (viz Obrázek 3). Jedná se o programovatelný logický kontroler (PLC) s integrovanými moduly (čítání, měření, regulace, řízení, ...), které tvoří všestranný řídicí systém.



Obrázek 3 PLC Siemens S7-1200

Jedná se o jeden z nejpopulárnějších PLC na trhu, a to především díky malým rozměrům, nízké spotřebě a vysoké flexibilitě. Další výhodou PLC Siemens S7-1200 je jeho snadná integrace s ostatními moduly v průmyslové automatizaci, mezi které patří

senzory, akční členy či HMI panely. Vzhledem k výkonu a vysoké flexibilitě patří tento modul mezi jedno z nejlepších řešení pro průmyslovou automatizaci v malých až středních firmách. Mezi přednosti modulu patří zabezpečení proti neoprávněnému vniknutí či zneužití dat (v dnešní době je třeba apelovat na zabezpečení a zneužití dat), jedná se o ochranu proti kopírování, neoprávněné otevírání nebo duplikování modulů. Pro jednoduchou a komplexní komunikaci mezi programátorem a modulem slouží TIA Portal.

Tabulka 1 Specifikace S7-1200

CPU	1516-3 PN/DP
Pracovní paměť	1 MB pro program, 5 MB pro data
Protokoly	TCP/IP, HTTP, HTTPS, SNMP, DCP, LLDP, S7 Routing
Ethernet	2X RJ-45

4 Software a použité technologie

4.1 TIA Portal

TIA Portal je komplexní vývojové prostředí od firmy Siemens. Mezi hlavní softwarové části patří tvorba uživatelských algoritmů a aplikací pro PLC, projektování uživatelských ovládacích panelů HMI, komunikačních komponent, řízení pohybu atd. Ve prospěch komplexního řešení v jednotném vývojovém prostředí se jedná o zjednodušení práce s jednotlivými nástroji. TIA Portal byl během vývoje použit pro vytváření testovacích proměnných, datových bloků a ovládání stanice (restart, nahrání potřebných souborů do paměti, zobrazení reálných hodnot stanice).

4.2 Ajax

Ajax (Asynchronous JavaScript and XML) je webová technologie používána pro interaktivní webové aplikace. Hojně využívána pro webové aplikace typu SPA (Single Page Application), kdy se všechno obsah a dotazy odehrávají v jedné stránce (v pozadí). Princip načítání obsahu stránky je řešen pomocí asynchronního zpracování, při kterém nedochází k obnovení celé stránky, ale pouze požadované části. Zjednodušeně, pokud uživatel potřebuje zobrazit konkrétní informace na stránce, přičemž se změní pouze část obsahu a zbylá část obsahu je neměnná, dojde pouze k načtení požadované informace, a nikoliv celé stránky. Pro uživatele je tak stránka přehlednější a celý proces obsluhy plynulejší.

4.3 JQuery

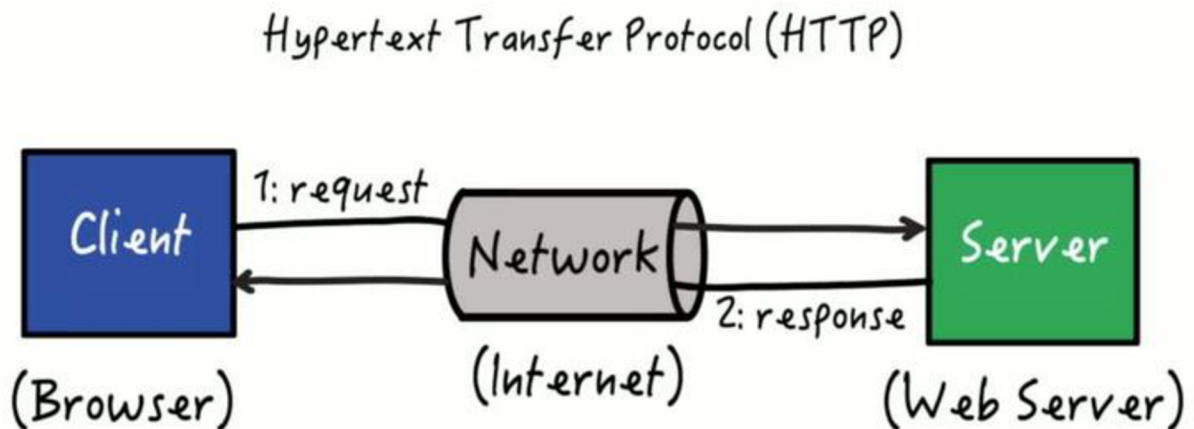
Jedná se o knihovnu, která usnadňuje práci v JS (JavaScript) a komunikaci s HTML. Mezi hlavní rysy užitečnosti patří manipulace s obsahem stránky, zpracování událostí a v neposlední řadě použití AJAX. Jedná se o nejpoužívanější a nejrozšířenější knihovnu pro JS a v použití s mnoha JQuery pluginy můžeme dosáhnout požadovaných interaktivních zlepšení (carousely, vylepšené formulářové prvky, ...).

4.4 Výměna dat mezi klientem a serverem

Výměna dat mezi PLC a webovým rozhraním zahrnuje přenos informací o aktuálním stavu dobíjecí stanice a prováděných operací. Jedná se o kritickou součást

celého procesu dobíjení baterií. Efektivní a spolehlivá komunikace mezi těmito systémy zajišťuje, že uživatel spravuje a monitoruje dobíjecí stanici s minimem rizik a chyb.

Samotný proces komunikace mezi klientem a serverem obsahuje několik kroků. Při použití protokolu http se klient (typicky je to webový prohlížeč) připojí k serveru a odešle na něj požadavek. Tento požadavek obsahuje informace a instrukce, které chce klient od serveru získat. Server požadavek zpracuje a následně odešle klientovi požadované informace. Odpovědí může být soubor dat v JSON či přímo webová stránka. Tyto data či webová stránka jsou pak zobrazeny uživateli (viz Obrázek 4). Obdobný proces probíhá při komunikaci přes HTTPS protokol, ale v této variantě je využito šifrování. Mezi klientem a serverem musí proběhnout tzv. „TLS Handshake“ (neboli proces navazování bezpečného spojení,) kde dojde k výměně klíčů pro šifraci a dešifraci dat. [1] Po navázání bezpečného spojení odešle klient požadavek na server, server odpověď zašifruje a odešle zpět na klienta. Klient přijatá data dešifruje pomocí stejného klíče a zobrazí data uživateli. I přesto, že se jedná o zabezpečený přenos citlivých dat, dochází stále častěji k útokům, a proto je nutné dbát na celkovou bezpečnost aplikace a co nejvíce zamezit možnosti vniknutí útočnickovi do systému. [4]



Obrázek 4 Schéma HTTP protokolu

4.5 Možnosti zabezpečení webové aplikace

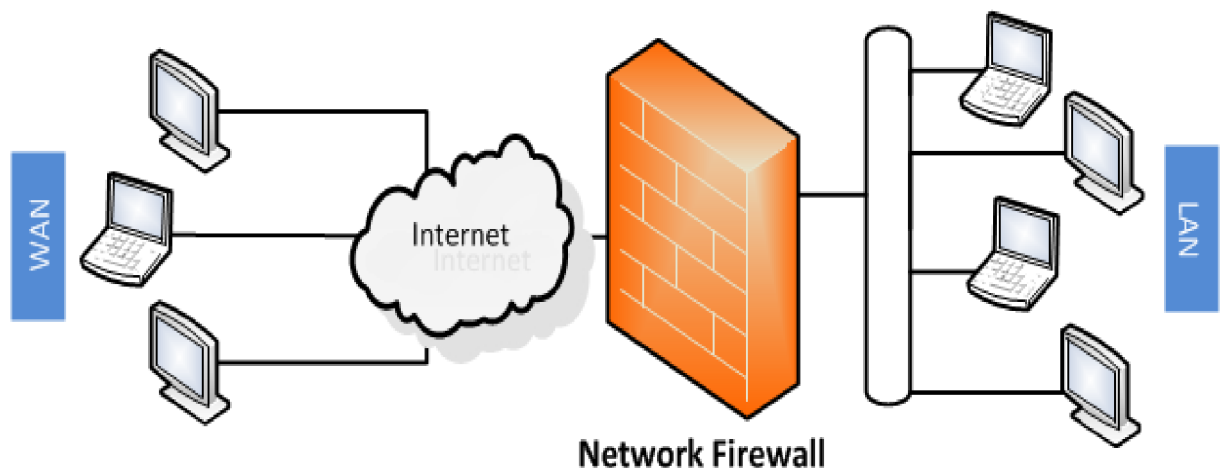
Při zabezpečení vždy musíme zohlednit celkový stav realizace (jak webové aplikace, tak ovládané stanice). V případě instalace dobíjecí stanice pro koncového zákazníka, kdy zákazník vlastní dobíjecí stanici pouze pro osobní užití, počítáme, že nároky na bezpečnost nebudou tak vysoké jako při použití veřejných dobíjecích stanic.

Pokud zvolíme možnost instalace dobíjecí stanice pro osobní použití, a ne komerční, můžeme zvolit jednoduché kroky pro zabezpečení přístupu a ovládní stanice. V případě fyzické ochrany je dobré mít samotnou stanici instalovanou na neveřejném místě (nejlépe v garáži). Tímto zabráníme k nechtěnému poškození cizí osobou či zneužití. Co se týče webové ochrany, je dobré implementovat přihlášení do aplikace pomocí hesla a uživatelského jména. Tímto snížíme riziko proniknutí do aplikace a zneužití. Při instalaci dobíjecí stanice pro osobní použití, uvažujeme připojení stanice do domácí lokální sítě (LAN). Tato síť je chráněna před okolním světem (WAN – wide area network) pomocí routeru. Samotná funkce routeru neposkytuje pouze přístup k internetu, ale také směrování, přenos datových toků a zabezpečení.

Samotný router poskytuje ochranu pomocí přihlášení (je vhodné změnit přihlašovací údaje na vlastní a nepoužívat implementované přihlašovací údaje od výrobce např. heslo: admin, přihlašovací jméno: admin). Dalším krokem pro ochranu je, aktualizovat interní firmware routeru (samotný firmware obsahuje implementaci zabezpečení, nové šifrovací metody, obranu proti útokům). Pokud vlastníme bezdrátovou síť, je vhodné nastavit dostatečně silné heslo a nepoužívat zabezpečení pro hesla typu WEP, které se považují za zastaralé a lehce prolomitelné.

Další možností ochrany je filtrování a omezení přístupu pro konkrétní zařízení pomocí identifikace MAC adresou (unikátní adresa každého zařízení). Většina nových routerů poskytuje možnost nastavení (omezení/zamítnutí přístupu k internetu) pro konkrétní zařízení, a tím můžeme povolit pouze adresy zařízení, které v domácnosti používáme a víme, že jsou bezpečné. V tomto případě je vhodné používat ochranný antivirový program v těchto zařízeních (mobil, pc, tablet), abychom eliminovali hrozbu napadení našeho zařízení a prolomení ochrany.

V poslední řadě mezi jednu z nejpoužívanějších ochran patří brána Firewall. [7] Jedná se o zařízení, které slouží jako hlavní prvek ochrany a sledování síťového provozu. Je to kontrolní bod, který definuje pravidla pro komunikaci mezi oddělenými sítěmi. V dnešní době Firewall kontroluje i kvalitu spojení, používané certifikáty a protokoly či prvky IDS (prvky, které sledují internetový provoz a detekují anomálie a případné hrozby a narušení bezpečnosti). V domácí lokální síti je tato implementace nezbytná, jelikož je ze všech zmíněných metod a implementací nejsilnější. Je mnoho druhů routerů, každý má jiné ochranné prvky a od toho se odvíjí cena zařízení, avšak v dnešní době je vhodné investovat do zabezpečení a chránit tak citlivá data či nechtěné zneužití ovládaného zařízení. [6]



Obrázek 5 Schéma Firewall brány

V případě komerční instalace dobíjecích stanic (sít' veřejných dobíjecích stanic) se apeluje na celkové zabezpečení, jak samotné dobíjecí stanice, tak i přístupové aplikace.

5 Vývoj a testování

5.1 Návrh aplikace

Při samotném návrhu bylo zapotřebí pečlivě rozvrhnout jednotlivé kroky práce. Prvním krokem bylo zjištění možností hardwarových komponent, následoval výběr vhodných knihoven pro usnadnění práce. Pro testování a vytváření jednotlivých testovacích proměnných byly použity TIA Portal od firmy Siemens a VS Code. Při návrhu grafického rozhraní byl kladen důraz na jednoduchost a na to, co uživatelé potřebují vidět a co je potřeba umožnit jim měnit. Estetičnost a snadnost ovládání byly důležité faktory, které vedly k vytvoření jednoduchého konceptu vizuální prezentace webové stránky. [11]

5.2 Řídící logika programu

Mezi nejdůležitější prvky webové aplikace patří část kódu, která se stará o předávání informací mezi systémem a uživatelem. Tato část je založena na architektuře klient-server.

Celá aplikace obsahuje dva základní bloky ovládací logiky. První část se stará o načítání hodnot do stránky, kdy jsou z PLC načteny aktuální hodnoty. Tyto hodnoty jsou následně zobrazeny uživateli v grafické podobě. Druhá část logiky se stará o zpracování zadaných hodnot od uživatele. Uživatel může měnit napájecí proud a kapacitu nabití baterie. Tyto změny jsou odeslány na server a následně zpracovány v PLC.

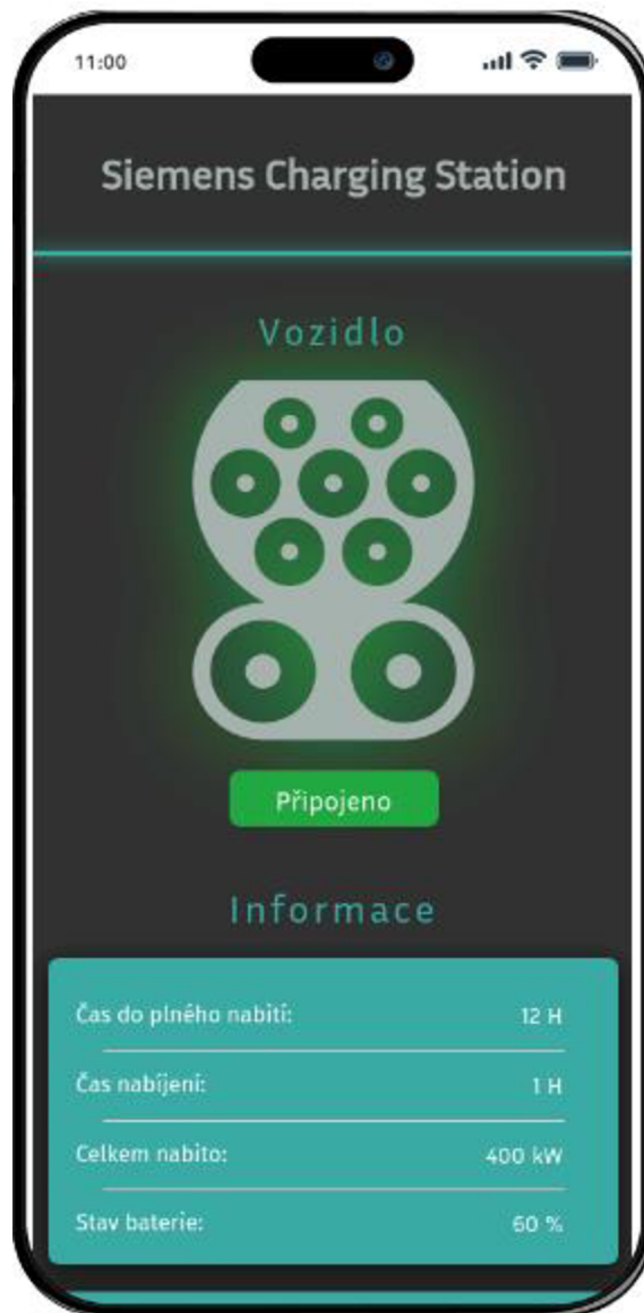
5.3 Výměna dat

Samotná komunikace obvykle probíhá za pomoci protokolu HTTP či HTTPS. Při každém požadavku, jako je například požadavek na zobrazení aktuálního výkonu nabíjení, webové rozhraní odešle žádost na PLC server. Server následně žádost zpracuje a odešle zpět potřebná data. Tato data jsou pak zpravidla ve formátech JSON nebo XAML a obsahují všechny požadované informace. Webové rozhraní následně data zpracuje a zobrazí uživateli na obrazovce.

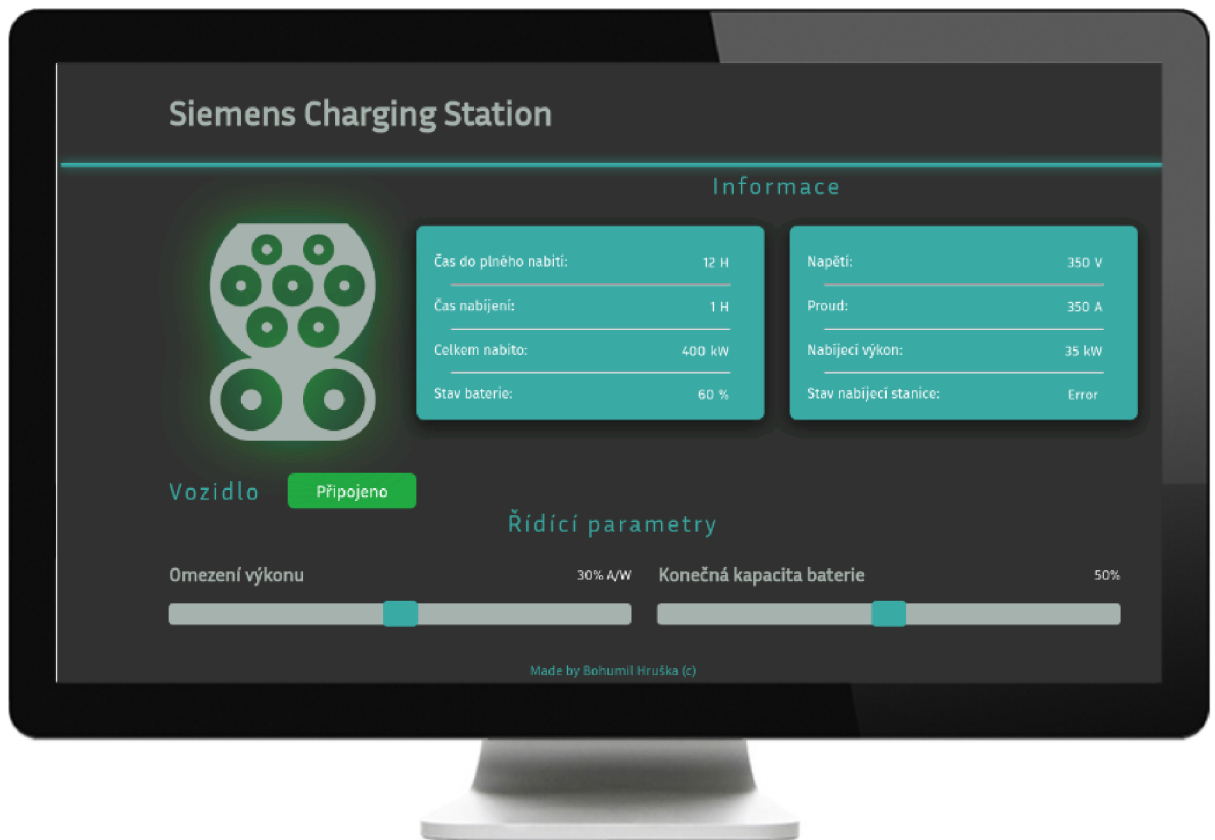
Obdobný způsob nastane, pokud uživatel požaduje změnit například maximální nabití baterie. Webové rozhraní pošle požadavek na PLC server s žádostí o změnu hodnoty interní proměnné, po dokončení operace odešle PLC informaci o zpracování požadavku zpět do webového rozhraní, a to následně zobrazí uživateli výsledek operace.

5.4 Grafické rozhraní

Grafická vizualizace je velice důležitá, pokud se jedná o ovládací program (webovou aplikaci v tomto případě), a proto při grafickém návrhu bylo dbáno na jednoduchost a efektivnost. Aplikace je navržena a testována především pro mobilní zařízení (viz Obrázek 6), avšak umožňuje zobrazení a ovládní pomocí jakéhokoliv zařízení připojeného k internetu (viz Obrázek 7), a to díky responsivnímu návrhu. Samotná vizuální stránka webová aplikace je psána značkovacím jazykem HTML a doplněna o stylový vzhled pomocí souboru CSS. Webová aplikace umožňuje uživateli zobrazovat stavové hodnoty nabíjecí stanice mezi které patří například indikace nabíječe (připojen – konektor modrý, nepřipojen – konektor šedý, chyba – konektor červený), zobrazení hodnot nabíjecí stanice (proud, čas do nabití, výkon, ...). Dále pak umožňuje uživateli ovládat hodnoty nabíjení jako je vstupní nabíjecí proud či změnu výkonu pomocí posuvníků v dolní části grafického rozhraní.



Obrázek 6 GUI – mobilní verze



Obrázek 7 GUI – desktopová verze

5.5 Testování webové aplikace

Testování a odchyťávání chyb patří k součásti vývoje aplikací, aby bylo dosaženo maximální efektivnosti a minimální chybovosti. Grafická prezentace webové aplikace byla testována během vývoje na různých zařízeních (pc, tablet, mobil) a také na různých prohlížečích (Chrome, Safari, Edge). Samotná tvorba grafické vizualizace proběhla mnohými změnami, a proto bylo testování náročnější. Při tvorbě řídicích metod (načtení/nahrání dat z/do PLC) se samotné funkce testovaly na fiktivních datech a virtuálním stroji tak, aby při nastání chyby nedošlo k poškození reálného stroje (nabíjecí stanice či automobilu). Po otestování dílčích metod se propojila logická vrstva s vrstvou grafickou a aplikace se začala testovat jako celistvý projekt. Při testování byly odhaleny a eliminovány chyby. U grafické stránky se jednalo o malý posuvník na mobilní verzi, který byl špatně ovladatelný na zařízeních s malou obrazovkou. Jako menší problémy vznikly odchylky barev pro různé prohlížeče (některé vady byly odstraněny a některé nebránily provozu a estetičnosti). U logické stránky došlo při testování k úpravě formátů některých proměnných nebo k úpravě intervalu pro načítání hodnot z PLC (PLC Siemens podporuje nejméně 500ms). Po celkovém otestování došlo k nasazení aplikace na reálnou dobíjecí stanici a postupně byla odzkoušena všechna funkcionalita.

6 Tvorba knihovny komponent

Knihovna komponent pro TIA Portal je soubor předpřipravených prvků, které můžete použít při tvorbě řídicího programu v programovacím prostředí TIA Portal. Tyto komponenty obsahují již naprogramované funkce, bloky a moduly, které mohou být snadno přizpůsobeny pro konkrétní aplikaci. [10]

Tyto knihovny komponent usnadňují tvorbu programů v TIA Portalu, a tím i zkracují celkový čas potřebný pro vývoj a nasazení řídicích systémů. Knihovny mohou obsahovat různé prvky, jako jsou například bloky pro řízení pohybu, komunikační bloky, vizualizační prvky, bloky pro regulaci a mnoho dalšího.

V TIA Portalu jsou k dispozici předpřipravené knihovny komponent od výrobce Siemens, ale mohou být také vytvářeny uživateli a sdíleny s ostatními uživateli.

6.1 Tvorba datového bloku

Pro zobrazení dat z PLC do webové aplikace je potřeba vytvořit datové bloky v TIA Portalu. Pomocí tohoto datového bloku se budou nahrávat hodnoty z PLC do webové aplikace. V datovém bloku se tvoří proměnné, které budeme následně využívat pro zobrazení skrze webovou aplikaci. V případě přípravy programu pro reálný hardware je důležité nejdříve program otestovat na virtuálním systému a po následném odhalení a odstranění chyb, odzkoušet program na reálném hardware. TIA Portál umožňuje vytvořit datové bloky pro testování a po otestování je možné napojit datové bloky reálného hardware přímo na námi vytvořené datové bloky (odpadá nutnost přejmenovávat všechny proměnné v samotném TIA Portálu, ale i kódu webové aplikace).

Samotné rozhraní umožňuje při vytváření proměnných (viz Tabulka 2) vybrat z předpřipravených datových typů. Při přenosu informací z PLC do webové aplikace může dojít k rozdílnému kódování znaků a je nutné výpis konkrétně použitých proměnných otestovat. V některých verzích docházelo k problému při přenosu textového řetězce (String). Při zobrazení na webové stránce docházelo k nahrazení znaku apostrofů, mezer, dvojteček a dalších za ascii znaky, které nebyly pro uživatele přívětivé.

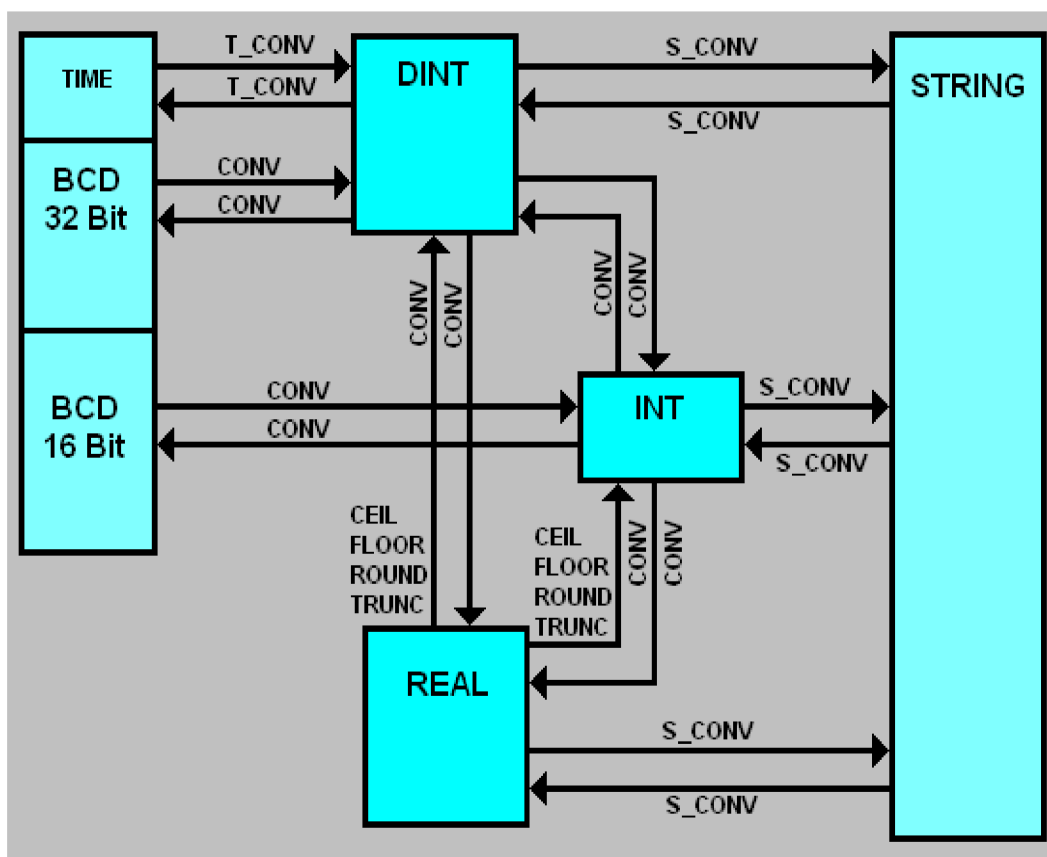
Tabulka 2 Seznam datových typů

Datový typ	Velikost v bitech	Příklad zobrazení STL
BOOL	1	TRUE/FALSE
BYTE	8	B#16#10
WORD	16	2#0001_0000_0000_0000
DWORD	32	2#1000_0001_0001_1000_ 1011_1011_0111_1111
INT	16	101
DINT	32	L#101
REAL	32	1.234567e+13
S5TIME	16	S5T#0H_1M_0S_0MS
TIME	32	T#0D_1H_1M_0S_0MS
DATE	16	D#1996-3-15
TIME_OF_DAY	32	TOD#1:10:3.3
CHAR	8	'E'
STRING	16	'HELLO'

6.2 Napojení datových bloků

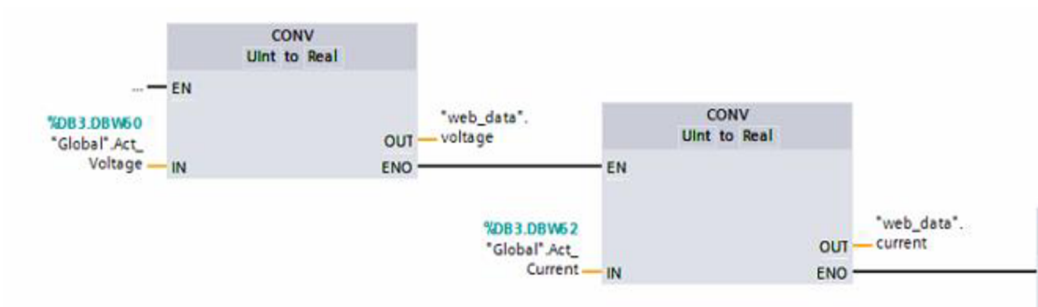
Pro otestování reálného systémů na námi vytvořené aplikaci a datových blocích můžeme napojit reálná data přímo na tyto datové bloky.

Při napojování datových bloků můžeme využít již vytvořených funkcí pro napojení. Na níže uvedeném obrázku (viz Obrázek 8) jsou popsány nejvíce používané typy konverzí (viz Obrázek 9) mezi datovými typy. Například pro převod u datového typu Real můžeme použít více funkcí (Ceil – zaokrouhlení na celé větší číslo, Floor – zaokrouhlení na celé nižší číslo, Round – zaokrouhlení na celé číslo, Trunc – bez zaokrouhlení). Pro konverzi textových řetězců (String) můžeme využít také více funkcí (STRG_VAL – převod textu na číslo, VAL_STRG – převod čísla na text, Char_TO_Strg – převod znaků na textový řetězec). Konkrétní použití funkcí záleží na vstupních a očekávaných výstupních datech. [12]



Obrázek 8 Nejpoužívanější funkce pro konverzi datových typů

Výsledná realizace by v TIA Portal mohla vypadat následovně:



Obrázek 9 Příklad použití funkce CONV

Z datových bloků reálného systému (v obrázku se jedná o datový blok „Global“) jsme napojili data na námi vytvořený datový blok s názvem „web_data“. Pro převod byla použita funkce CONV, dochází zde k převodu UInt na Real. Na levé straně převodníku je definována proměnná „Act_Voltage“ z reálného systému a na pravé straně proměnná „voltage“ z našeho virtuálního systému, který jsme použili pro otestování funkčnosti. Tímto převodem dojde k předání hodnot z datového bloku „Global“ do datového bloku „web_data“ a tím odpadá potřeba přejmenovávat název datového bloku a proměnných přímo v kódu. Jedná se tedy pouze o propisání hodnot mezi datovými bloky. V pravé části obrázku můžeme vidět obdobný proces pro proměnnou „current“.

V případě, že datový typ v datovém bloku s reálnými hodnotami odpovídá datovému typu v námi vytvořeném datovém bloku můžeme použít funkci MOVE, která pouze propisuje hodnotu z jednoho datového bloku do druhého. V obrázku níže pouze přiřazujeme hodnotu 10 na adresu „Tag_1“. Hodnotu 10 je možné nahradit jinou proměnnou z jiného datového bloku, přičemž musí být zachována stejnost datových typů na obou stranách funkce MOVE.



Obrázek 10 Příklad použití funkce MOVE

7 Metodika tvorby vlastních stránek

K nahrání obsahu na webové stránky lze využít již předpřipravených šablon od výrobce nebo vytvoření vlastního obsahu stránek. Nicméně pokud uživatelé hledají větší flexibilitu a kontrolu nad obsahem a designem webových stránek, mohou vytvořit své vlastní stránky.

Tato příručka popisuje základní práci a postup při tvorbě vlastních uživatelských stránek (User-Defined Web Pages) [9], což umožňuje uživatelům vytvořit obsah a design podle svých konkrétních potřeb. Při samotné tvorbě vlastních stránek je důležité brát v potaz, že konečný design a obsah stránek by měly být snadno srozumitelné pro koncového uživatele webu. Důležité je také zajistit, aby webová aplikace byla responsivní a přizpůsobivá na různých zařízeních (mobilní telefony, tablety, ...). Kromě toho by tvůrci měli myslet na to, že vytváření uživatelských stránek vyžaduje určité znalosti jazyka HTML, CSS, JS a jiných webových technologií.

Celkově lze říci, že tvorba vlastních uživatelských stránek může být náročná, ale zároveň přináší mnoho výhod v podobě plné kontroly nad designem a funkcionalitou webových stránek. Je důležité mít na paměti všechny faktory, které by mohly ovlivnit výsledný produkt, a zajistit, aby webové stránky byly snadno srozumitelné, responsivní, dobře zabezpečené a optimalizované pro vyhledávače.

7.1 Inicializace HTML dokumentu

V první řadě je třeba do hlavičky stránek (tag `<head>`) vložit potřebné knihovny pro práci. V našem případě se jedná o knihovnu JQuery a tu vložíme do hlavičky následovně:

`<script src="jquery-3.6.3.min.js"></script>`. Je možné importovat více knihoven (ne všechny jsou interním web serverem podporovány). Například pro tvorbu responsivního designu stránek můžeme využít knihovny Bootstrap. Konečný obsah hlavičky by mohl vypadat takto.

```
<head>
  <meta charset="utf-8"> <!--Typ kódování dokumentu-->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> <!--Nastavení zobrazení pro mobily-->
  <meta name="description" content="BP"> <!--Název práce-->
  <meta name="author" content="Bohumil Hruška"><!--Autor-->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
  <link rel="stylesheet" href="style.css" type="text/css"> <!--Odkaz na vlastní css styly-->
</head>
```

Dále můžeme pokračovat v psaní HTML dokumentu jako při tvorbě klasické stránky. Vytvoříme ovládací a zobrazovací prvky a formuláře, do kterých budeme zobrazovat hodnoty. V případě tvorby logiky (Javascript, JQuery, ...) v externím souboru, kdy nebudeme psát logiku programu přímo do .html souboru, je nutné do hlavičky přidat odkaz na tento soubor .js, který obsahuje logické funkce.

7.2 Vytvoření datových bloků

Datové bloky je možné si představit jako místo v paměti, kde jsou uloženy proměnné pro vývoj. V datových blocích vytváříme proměnné, které budeme později zobrazovat uživateli. Nový datový blok vytvoříme pomocí **Add new block** v stromovém adresáři projektu v podsložce **Program blocks**. Zobrazí se nám dialogové okno s výběrem, jaký blok chceme vytvořit (Organization block, Function block, Function, Data block). V našem případě se jedná o Data block. Popis jednotlivých bloků je popsán níže.

Organization block – výběr z více možností, jedná se o cyklické funkce, přerušení.

Function block – jedná se o blokové kódy, které uchovávají data permanentně jako instance datového bloku, je možné s nimi pracovat i po spuštění programu.

Function – blok kódu, který nemá vyhrazenou paměť.

Data block – slouží k uložení dat programu.

7.3 Vytvoření proměnných

Po vytvoření datového bloku můžeme přejít na vytvoření proměnných, které později budeme zobrazovat na webové stránce. Vybereme datový blok, ve kterém chceme proměnnou vytvořit. Pomocí tlačítka **Insert row** nebo **Add row**. Zadáme název proměnné, datový typ (int, float, string, DateTime, ...), počáteční hodnotu a zbylé parametry necháme automaticky. Během testování je také možné sledovat aktuální hodnotu proměnné za pomoci tlačítka **Monitor all** (najdeme jej na ovládací liště vedle tlačítek pro vytvoření proměnné). [3] [2]

7.4 Čtení a zápis proměnných

Při definování proměnné používáme dvě metody. První metodou lze pouze číst hodnotu proměnné a tento zápis se píše přímo do html elementu (text, odstavec, buňka tabulky, ...). Zobrazení proměnné **cisloA** v odstavci by vypadalo následovně.

```
<p>Číslo A se rovná: :="<cisloA>":</p>
```

Druhá metoda nám umožňuje předem definovat proměnnou a dále ji pak využít k zápisu a čtení. Tento zápis je nutný pro práci se speciálními proměnnými. Obecný zápis:

```
<!-- AWP_In_Variable Name=""DB".název_proměnné' -->.
```

V zápisu se objevuje **DB (název datového bloku)**, který souvisí s umístěním proměnné. Níže si vysvětlíme různé typy zápisu podle umístění proměnné.

7.5 Čtení a zápis z DB www stránek

Při zápisu či čtení proměnných se používá speciálního zápisu. Vždy záleží na místě uložení proměnné. Pokud potřebujeme číst/zapisovat přímo z bloku webové stránky (tag table) definujeme proměnnou pouze pomocí jména a to následovně: :=**“jméno_proměnné“**:. V praxi to pro nás znamená, že pokud máme proměnnou s názvem **cisloA** a budeme ji chtít zobrazit v odstavci, tak její zápis v tomto případě bude **<p>:=“cisloA“:</p>**.

7.6 Čtení a zápis z libovolného DB

V případě, kdy potřebujeme načítat či zapisovat hodnoty (viz Tabulka 3) z jiných datových bloků (mají umístění jinde než v DB web) musíme proměnnou definovat i s názvem datového bloku. Obecný zápis:

```
<!-- AWP_In_Variable Name=""DB".název_proměnné' -->.
```

Zápis proměnné **cisloA** uložené v DB jménem **data** <!-- AWP_In_Variable Name=""data".cisloA' -->.

Tabulka 3 Přehled zápisu proměnných

Obecný zápis	Konkrétní zápis	Popis
:=“Proměnná“:	<p>:=“voltage“:</p>	Zobrazení proměnné, která je definována jako globální (není nutné zadat název datového bloku).
:=“DB ".Proměnná:	<p>:=“web_data".voltage:</p>	Zobrazení proměnné voltage , která je umístěna v datovém bloku web_data .
<!--AWP_In_Variable Name=""DB".název_proměnné' -->	<!--AWP_In_Variable Name=""web_data".voltage' -->	Definování proměnné voltage , která je umístěna v datovém bloku web_data . Tato deklarace slouží pro povolení přenosu hodnot do PLC pomocí metody POST.

7.7 Zobrazení dat

Při zobrazování dat z PLC do obsahu webové stránky můžeme využít dva různé způsoby. První způsob získání dat z PLC, který je jednodušší se nazývá „Statické zobrazení“ dat. Tento způsob spočívá v načtení dat z PLC v jednom okamžiku a následném zobrazení těchto dat. Statické zobrazení je vhodné pro případy, kdy potřebujeme získat interní nastavení nebo načíst neměnné nebo hodnoty s malou frekvencí změny.

Druhý způsob, který se nazývá Dynamické zobrazení dat, se používá, pokud potřebujeme zobrazovat aktuální data (Live data), která se mohou měnit a která potřebujeme monitorovat. Dynamické zobrazení umožňuje okamžitou reakci na změnu stavu hodnoty v PLC. Dynamické zobrazení se nejčastěji používá v průmyslových aplikacích, kde je důležité monitorovat různé parametry a celkový stav zařízení v reálném čase.

Při použití dynamického zobrazení je obvykle použita technologie AJAX, která je popsána v kapitole 44.2. Dále je vhodné zajistit, aby se data z PLC načítala v co nejkratším čase, aby uživatelé viděli aktuální data v reálném čase. Kromě toho je důležité zajistit, aby webová aplikace byla dobře optimalizována a minimalizovala se zátěž na server (zejména při načítání velkého množství dat).

Volba druhu aplikace Statického zobrazení či Dynamického, vždy záleží na typu aplikace a na konkrétních požadavcích realizace.

Příklad statického zobrazení

Meteorologická stanice pro zobrazení aktuální teploty a vlhkosti. V tomto případě uživatel aplikaci použije párkrát za den a chce vidět aktuální data teploty a vlhkosti. Při těchto datech se nepočítá, že by se jejich aktuální hodnota měnila rapidně oproti teplotě při spuštění aplikace, a proto není potřeba využít dynamického zobrazení.

Příklad dynamického zobrazení

Stejný příklad jako výše uvedený, ale jedná se o stanici, která ukazuje aktuální rychlost větru během střelby na terč. V takovém případě je vhodné použít dynamické zobrazení tak, aby střelci měli k dispozici aktuální rychlost větru.

7.8 Statické zobrazení dat

Zobrazit data na stránce můžeme pomocí HTML tagu například v tagu <p> nebo pomocí Javascriptu a knihovny jQuery. Při otevření webové stránky se nám zobrazí hodnoty jednotlivých proměnných (musí být definované v DB s názvem „web_data“). Zde můžeme vidět výše popsaný zápis proměnných. Pokud se změní hodnoty v PLC a budeme chtít tyto hodnoty vidět, musíme aktualizovat (obnovit) webovou stránku, aby došlo k opětovnému načtení správných hodnot do webové stránky.

```
<div role="main" class="ui-content">
  <p>Data načtená z PLC</p>
  <p></p>
  <p>
    DI 0: :="web_data".DI0:<br>
-->
    DI 1: :="web_data".DI1:<br>
-->
    DI 2: :="web_data".DI2:<br>
-->
    DI 3: :="web_data".DI3:
  </p>
  <p>
    AI 0: :="web_data".AI0:<br>
-->
    AI 1: :="web_data".AI1:
  </p>
</div>
```

!<!-- Tělo stránky -->
!<!-- Text na stránce -->
!<!-- Nový řádek -->
!<!-- Nový odstavec hodnot -->
!<!-- Hodnota DI 0 a odskok na nový řádek -->
!<!-- Hodnota DI 1 a odskok na nový řádek -->
!<!-- Hodnota DI 2 a odskok na nový řádek -->
!<!-- Hodnota DI 3 -->
!<!-- Hodnota AI 0 a odskok na nový řádek -->
!<!-- Hodnota AI 1 -->

Obdobný výsledek dostaneme, pokud použijeme JavaScript a jQuery. Vytvoříme jednoduchou funkci pomocí jQuery (ošetřená manipulace se stránkou je vhodná až po úplném načtení, jQuery zjistí, kdy lze se stránkou bezpečně operovat a vykoná námi definovanou funkci).

Do proměnných začínajících slovem „var“ nahrajeme hodnoty načtené z PLC a tyto proměnné použijeme dále v kódu (například pro výpis do <p> elementu). V případě, kdy bychom chtěli nejen zobrazovat hodnoty, ale také například upravovat hodnotu některé proměnné je nutné na začátek celého .html souboru vložit řádky, které povolují zápis do proměnné. Například budeme mít v DB proměnnou „current“ a na webové stránce budeme chtít upravovat její hodnotu pomocí Javascriptu. V tomto případě musíme na začátek dokumentu vložit **<!--AWP_In_Variable Name=""web_data".current' -->**.

```
<script>
$(document).ready(function(){
  var hodnota_DI0 = :="web_data".DI0;;
  var hodnota_DI1 = :="web_data".DI1;;
  var hodnota_DI2 = :="web_data".DI2;;
  var hodnota_DI3 = :="web_data".DI3;;
  var hodnota_AI0 = :="web_data".AI0;;
  var hodnota_AI1 = :="web_data".AI1;;
  $('#AI0').val(hodnota_AI0).slider("refresh");
  $('#AI1').val(hodnota_AI1).slider("refresh");
  console.log("Hodnoty načteny");
})
</script>
```

Pokud se rozhodneme upravovat proměnné, je vhodné přistoupit k implementaci dynamického zobrazení dat a asynchronního načítání dat pomocí AJAX volání. Pro statické zobrazení není vhodné upravovat hodnoty proměnných, jelikož by uživatel neviděl změnu, dokud by neobnovil celou stránku. Při změně hodnoty proměnné je nutné, aby uživatel, který aplikaci ovládá, dostal zpětnou reakci o tom, že změna proběhla úspěšně či ne.

7.9 Dynamické zobrazení dat

Níže popsaná funkce update slouží k načítání hodnot z PLC a zároveň k posílání upravených hodnot z posuvníků (rangePower, rangeBattCapacity). Načtené hodnoty jsou v podobném formátu jako .csv soubory. Je nutné je tedy oddělit na jednotlivé hodnoty pomocí funkce split() a trim(). Jednotlivé hodnoty pole pak předáme do funkce updateValue(). Tato funkce přijímá 2 parametry – id html elementu, kde bude provedena změna hodnoty a samotnou novou hodnotu načtenou z PLC. Pokud by v přenosu nastala chyba, dojde k vyvolání výjimky a vypsání chybové hlášky do konzole.

```
function update() {
    powerReduction = $("#rangePower").val();
    maxBattCapacity = $("#rangeBattCapacity").val();

    try {
        $.get('data_in.html?randval=' + Math.random() + '&web_data.powerReduction='
+ powerReduction + '&web_data.maxBattCapacity=' + maxBattCapacity, function
(data) {
            var newData = data.trim().split(",");
            updateValue("timeToFullyCharged", newData[0]);
            updateValue("chargingTime", newData[1]);
            updateValue("totallyCharged", newData[2]);
            updateValue("battCapacity", newData[3]);
            updateValue("voltage", newData[4]);
            updateValue("current", newData[5]);
            updateValue("chargingPower", newData[6]);
            updateValue("stationStatus", newData[7]);
            SetchargerStatus(newData[8]);
        });
    }
    catch (err) {
        console.log("Chyba:" + err.message);
    }
}
```

8 Závěr

Bakalářská práce, kterou jsem vypracoval, nabízí nejen jedinečný přístup, ale také praktickou aplikaci v oblasti moderních technologií. Mým cílem bylo vytvořit inovativní webovou aplikaci pro ovládání automobilových dobíjecích stanic, a také sdílet své znalosti a zkušenosti v metodické příručce pro začátečníky v programování vlastních webových stránek pro PLC.

Samotná webová aplikace byla navržena s cílem vyhovět požadavkům společnosti Siemens. Při vývoji jsem postupoval podle ověřených metod pro vývoj aplikací. Vytvořil jsem základní koncept aplikace, od kterého se začalo vše odvíjet. Přes vývoj jednotlivých funkcí logiky, dynamického zobrazení live dat, vytvoření responsivního designu až po testování na virtuálním hardware. Výsledné řešení webové aplikace jsem testoval na mnoha zařízeních a prohlížečích, aby došlo k optimalizaci zobrazení a ovladatelnosti. Po procesu testování přišlo na řadu ověřit celou aplikaci na reálné nabíjecí stanici. Při vývoji aplikace a testování, jsem spolupracoval a konzultoval řešení s firmou Siemens a tím jsme docílili oboustranné spokojenosti.

Webovou aplikaci bych hodnotil jako perspektivní řešení, které má potenciál přinést významnou úsporu času a zvýšení efektivity provozu. Metodická příručka obsahuje detailní popisy, které nejsou výrobcem jasně popsány, a proto se tato příručka jeví jako hodnotný pomocník při tvorbě složitějších vlastních stránek pro PLC.

Dalším cílem bylo vytvořit metodickou příručku pro tvorbu vlastních stránek, která obsahuje detailní popis vytváření proměnných a následné použití na webové stránce, práci v TIA portálu a propojení PLC s webovou aplikací.

V budoucnu by webová aplikace mohla sloužit jako uživatelská aplikace pro správu nabíjení automobilů. Mezi první úpravy by bylo možné zahrnout proces autentizace pro zlepšení zabezpečení a neoprávněnému vniknutí uživatelů. Dalším vhodným rozšířením je vytvořit administrativní menu se statistickými údaji o nabíjení – plánování a sledování nabíjení, statistiky využití či ekonomická spotřeba nabíjení.

9 Seznam použité literatury

- [1] HTTP (Hypertext Transfer Protocol). TechTarget [online]. 2021 [cit. 2023-04-11]. Dostupné z: <https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol>
- [2] Creating and using user-defined web pages on S7-1200 / S7-1500 [online]. 2020 [cit. 2023-05-14]. Dostupné z: https://support.industry.siemens.com/cs/attachments/68011496/68011496_html_basics_for_simatic_cpus_en.pdf
- [3] Examples for the webserver of SIMATIC S7-1200 [online]. 2020 [cit. 2023-05-14]. Dostupné z: https://support.industry.siemens.com/cs/attachments/68011496/68011496_Examples_for_S7WebServer_DOC_v3_en.pdf
- [4] What is Web Application Security? [online]. 2020 [cit. 2023-05-14]. Dostupné z: <https://www.cloudflare.com/learning/security/what-is-web-application-security/>
- [5] Responsive Testing [online]. 2020 [cit. 2023-05-14]. Dostupné z: <https://www.lambdatest.com/learning-hub/responsive-testing>
- [6] The Hypertext Transfer Protocol (HTTP). In: Researchhubs [online]. 2015 [cit. 2023-05-14]. Dostupné z: <https://researchhubs.com/post/computing/web-application/the-hypertext-transfer-protocol-http.html>
- [7] Network Firewalls. In: We Solve It [online]. 2015 [cit. 2023-05-14]. Dostupné z: <https://www.wesolveit.co.uk/services/data-security/network-security-management/firewall-unified-threat-management/network-firewalls>
- [8] Obrázek PLC Siemens S7-1200. In: We Solve It [online]. [cit. 2023-05-14]. Dostupné z: https://www.plc-city.com/shop/435-thickbox_default/6es7212-1be40-0xb0.jpg
- [9] Creating and using user-defined web pages on S7-1200 / S7-1500: ID: 68011496. Industry Support Siemens [online]. Siemens, 2020, 04/07/2020 [cit. 2022-10-17]. Dostupné z: <https://support.industry.siemens.com/cs/document/68011496/creating-and-using-user-defined-web-pages-on-s7-1200-s7-1500?dti=0&lc=en-SK>
- [10] Learn-/Training Document. In: Siemens [online]. [cit. 2023-05-14]. Dostupné z: <https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/basics-programming-s7-1200/sce-031-600-global-data-blocks-s7-1200-r1709-en.pdf>
- [11] Fáze výroby stránek. In: Voxcafe [online]. [cit. 2023-05-14]. Dostupné z: <https://www.voxcafe.cz/blog/vyvoj-aplikaci/faze-vyroby-stranek.html>
- [12] In STEP 7 (TIA Portal), how do you convert the data types for the S7-1200/S7-1500?. Siemens [online]. 2016 [cit. 2023-05-14]. Dostupné z: [https://support.industry.siemens.com/cs/document/48711306/in-step-7-\(tia-portal\)-how-do-you-convert-the-data-types-for-the-s7-1200-s7-1500-?dti=0&lc=en-CZ](https://support.industry.siemens.com/cs/document/48711306/in-step-7-(tia-portal)-how-do-you-convert-the-data-types-for-the-s7-1200-s7-1500-?dti=0&lc=en-CZ)
- [13] Obrázek IOT [online]. [cit. 2023-05-18]. Dostupné z: <https://www.hw-group.com/cs/tiskove-zpravy/iot-proc-pouzit-ethernet>