

Monitorování a vyhodnocování bezpečnostních událostí z PLC

Bc. David Fiala

Diplomová práce
2016



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2015/2016

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David Fiala**
Osobní číslo: **A14392**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**
Forma studia: **prezenční**

Téma práce: **Monitorování a vyhodnocování bezpečnostních událostí z PLC**
Téma anglicky: **Monitoring and Evaluating Safety Events from a PLC**

Zásady pro vypracování:

1. Zpracujte literární rešerši problematiky týkající se možností programovatelných automatů (PLC) firmy Siemens.
2. Popište svařovací buňku a její propojení s PLC.
3. Navrhněte způsob získání, uložení a vyhodnocení požadovaných dat z PLC v závislosti na aktivních bezpečnostních událostech. Realizujte navržené úpravy v PLC programu.
4. Ověřte možnosti využití SCADA systému, případně webového serveru daného PLC, a to nejen z hlediska vizualizace daného procesu.
5. Ověřte navržené a realizované úpravy na svařovací lince a vyhodnoťte je.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **BOLTON, William.** Programmable logic controllers. 5th ed. Boston: Newnes, 2009, xii, 400 p. ISBN 978-185-6177-511.
2. **CENDELÍN, Jiří.** Historie programovatelných automatů a jejich současné efektivní použití. Automa [online]. 2003, 2003(6), 1 [cit. 2016-01-27]. Dostupné z: <http://automa.cz/historie-programovatelnych-automatu-a-jejich-soucasne-efektivni-pouziti-28831.html>
3. **KOVÁŘ, Josef a Ladislav ŠMEJKAL.** Programování PLC [online]. 1. Zlín, 2008 [cit. 2016-01-27]. Dostupné z: http://www.spszl.cz/soubory/plc/programovani_plc.pdf
4. **KUBÁŇ, Vojtěch.** Nová generace bezpečnostních automatů: Simatic S7-1500F. Automa [online]. 2014, 2014(6), 2 [cit. 2016-01-27]. Dostupné z: <http://automa.cz/res/pdf/52430.pdf>
5. **SIEMENS.** Technical data: 6ES7516-3FN01-OAB0 [online]. In: . s. 1 [cit. 2016-01-27]. Dostupné z: <https://support.industry.siemens.com/cs/pd/578299?pti=td&lc=en-WW>
6. **SIEMENS.** SIMATIC S7-1500: Getting Started [online]. 2014, 2014(5), 126 [cit. 2016-01-27]. Dostupné z: http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/software_complete_en.pdf
7. **ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ.** PLC a automatizace. 1. vyd. Praha: BEN technická literatura, 1999, 223 s. ISBN 80-860-5658-9.

Vedoucí diplomové práce:

Ing. Pavel Navrátil, Ph.D.

Ústav automatizace a řídicí techniky

Konzultant:

Bc. Tomáš Beck

Datum zadání diplomové práce:

19. února 2016

Termín odevzdání diplomové práce:

25. května 2016

Ve Zlíně dne 19. února 2016



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení: Bc. David Fiala

Název diplomové práce: Monitorování a vyhodnocování bezpečnostních událostí z PLC


Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně, dne


podpis diplomanta

ABSTRAKT

Práce se zabývá monitorováním a vyhodnocováním bezpečnostních událostí svařovací buňky, která je součástí svařovacího systému firmy AWL-Techniek, s využitím PLC značky Siemens.

Teoretická část obsahuje obecnému popisu programovatelných automatů, automatizačním prostředkům a vývojovému prostředí firmy Siemens. Dále popisuje svařovací buňku, její ovládací prvky, vybavení a propojení mezi jednotlivými prvky.

Praktická část popisuje datové bloky původního programu a všechny vytvořené funkční i datové bloky a zdrojové soubory webové stránky. Nakonec je vysvětlen účel výstupních zobrazení a vyhodnocení navržených změn.

Klíčová slova: PLC, programování, bezpečnostní událost, svařovací buňka, vizualizace

ABSTRACT

Thesis deals with the monitoring and evaluating of safety events of welding cell, which is a part of the welding system from company AWL-Techniek. It is used PLC from Siemens.

Theoretical part contains general description of programmable logic controllers, automation means and development environment of the company Siemens. Further it is described welding cell, control elements, equipment and connection between them.

Practical part describes data blocks from original program and all created data and function blocks and website source files. At the end there is explained purpose of the final visualizations and evaluation of the proposed changes.

Keywords: PLC, programming, safety events, welding cell, visualization

Tímto bych chtěl poděkovat svému vedoucímu práce, panu Pavlu Navrátilovi za neustále podněty při zpracování, za kritiku i osobitý přístup. Poděkování patří i firmě AWL za půjčení potřebného vybavení pro realizaci práce a především konzultantovi Tomáši Beckovi za zodpovězení spousty otázek.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY	11
1.1 OBECNÝ POPIS PLC.....	11
1.1.1 Hardware	11
1.1.2 Zpracování programu	12
1.1.3 Programování	13
1.2 AUTOMATIZAČNÍ PROSTŘEDKY ZNAČKY SIEMENS	14
1.2.1 PLC SIMATIC S7-1500F, CPU 1516F-3 PN/DP.....	14
1.2.2 Rozšiřovací moduly pro SIMATIC S7-1500F.....	16
1.2.3 Ostatní produkty firmy Siemens a označení	18
1.3 TIA PORTÁL.....	19
1.3.1 Nový projekt.....	19
1.3.2 Pracovní prostředí	20
1.3.3 Tvorba programu.....	21
1.3.4 Webserver	22
2 POPIS SVAŘOVACÍ BUŇKY	24
2.1 VNĚJŠÍ OVLÁDACÍ PRVKY	25
2.1.1 Bezpečnostní prvky	25
2.1.2 Human Machine Interface.....	26
2.2 VYBAVENÍ BUŇKY.....	28
2.2.1 Seznam zařízení	28
2.2.2 Propojení	30
II PRAKTICKÁ ČÁST	34
3 POPIS PROGRAMU	35
3.1 PŮVODNÍ PROGRAM.....	35
3.1.1 dbHmiSafetyMessages_1095	36
3.1.2 dbContext_71	39
3.1.3 dbMailbox_102	39
3.1.4 dbNameStrings_26.....	41
3.2 VYTVOŘENÝ PROGRAM	41
3.2.1 fbEventsLogging a dbfbEventsLogging	41
3.2.2 dbStatusBlock	42
3.2.3 fbWrite a dbfbWrite	43
3.2.4 fbWebServer a dbfbWebServer	46
3.2.5 fbEventsStats a dbfbEventsStats	49
3.2.6 fbWorkWithCsv a dbfbWorkWithCsv.....	50
3.2.7 Zdrojové soubory webové stránky	55
3.3 VÝSTUPNÍ ZOBRAZENÍ.....	57
3.3.1 CSV soubor	57
3.3.2 Webová stránka	58
3.3.3 Ukázka dat.....	61
4 VYHODNOCENÍ NAVRŽENÝCH A REALIZOVANÝCH ZMĚN	64

4.1	INTEGRACE DO STÁVAJÍCÍHO PROGRAMU	64
4.2	MOŽNÁ ROZŠÍŘENÍ	66
ZÁVĚR		68
SEZNAM POUŽITÉ LITERATURY		69
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		71
SEZNAM OBRÁZKŮ		73
SEZNAM TABULEK		75
SEZNAM PŘÍLOH		76

ÚVOD

Nedílnou součástí oboru dnešní automatizace jsou zařízení zvané programovatelné automaty. Tyto zařízení zkráceně nazývané PLC (*Programmable Logic Controller*) jsou schopné řídit celý výrobní proces bez ohledu na charakter výroby. Zároveň jsou schopny proces diagnostikovat, zálohovat data a v reálném čase je vizualizovat. Jeden z charakterů výroby může být svařování, který je neodmyslitelnou součástí nejen automobilového průmyslu. Programy vytvořené pro řízení procesů se také neustále vyvíjí nebo rozšiřují, aby vylepšili původní program nebo přidali další funkce a ulehčili práci obsluze. Právě tohle vylepšení je případ této práce.

Výsledky této práce by měly být využity holandskou firmou AWL-Techniek mající pobočku ve městě Napajedla. Firma se zabývá výrobou svařovacích buněk pro sériovou výrobu většinou pro automobilový průmysl. Tyto buňky jsou řízeny pomocí PLC, které při chybách vysílá bezpečnostní události. Některé z těchto události mohou zaniknout sami nebo po potvrzení na operátorském panelu a poté je teprve možno pokračovat ve výrobním procesu. V takovém případě je už událost a aktuální stav buňky nedohledatelný. Některé události už neumožní další chod buňky. Servisní technik, který poté chce odstranit závadu, která se objevuje například jen v určitých případech, musí simulovat události, aby zjistil, proč k tomu došlo, co zařízení uvnitř buňky právě dělaly, tedy jaký měly stav.

Vytvořením programu, který tyto události monitoruje a zaznamenává stav buňky a jednotlivých zařízení by se měla zkrátit doba pro identifikaci chyby a tím pádem by se měli zkrátit i prostoje při výrobě. Pro výstup programu, tedy vizualizaci byl zvolen CSV soubor a webová stránka. Obě varianty jsou dostupné přes webserver PLC, takže technik nepotřebuje dodatečný software. Vlastní vytvořený program pro PLC je snadno integrovatelný do stávajícího programu a funkčně ho rozšiřuje.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

1.1 Obecný popis PLC

Programovatelný logický automat, označován jako PLC je řídicí systém pro průmyslové a technologické procesy v reálném čase. Je uživatelsky programovatelný pro daný proces.

Jedná se o nástupce reléové techniky a univerzálních počítačů, které byly zbytečně drahé a složité pro potřebnou aplikaci. Proto v 70. začal nástup programovatelných automatů. Dříve byly určeny spíše pro úlohy logického typu, dnes zvládají regulační úlohy, monitorování procesů, či analogové měření.

Předností je rychlá realizace systému, kde stačí navrhnout vhodnou sestavu modulů pro daný proces, napsat uživatelský program a aplikovat návrh na reálný systém. Mezi největší výhody patří spolehlivost a robustnost zařízení i v horších podmínkách jako jsou zvýšené vibrace a vlhkost, nízké nebo naopak vysoké teploty a hluk. Zařízení je odolné proti rušení a poruchám. Oproti tomu PC je optimalizováno na početní úlohy a grafické zobrazování.

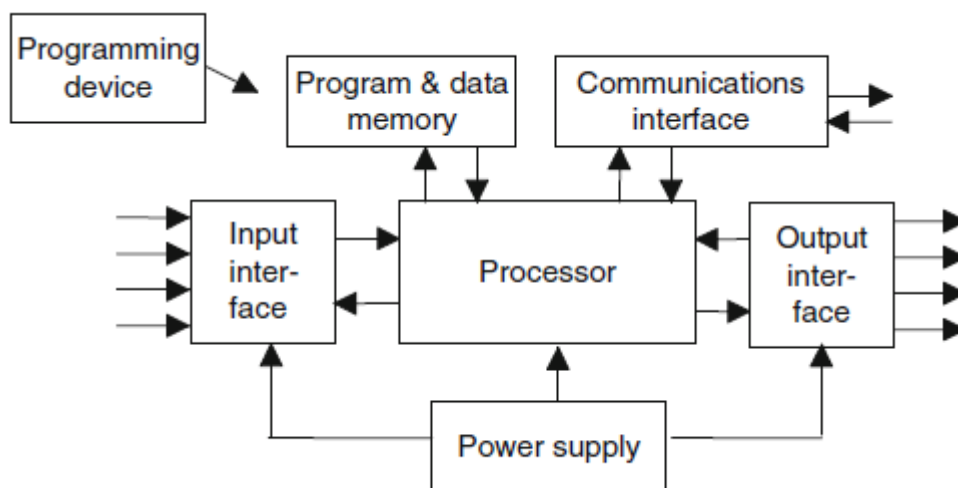
Výkonnost PLC se posuzuje podle doby potřebné pro vykonání instrukce. Tato doba se pohybuje v řádu nanosekund až mikrosekund. Tyto údaje odpovídají nejzákladnějším logickým instrukcím, ostatní trvají déle, proto lze objektivně posuzovat výkonost daného PLC na reálném příkladu, či typových úlohách. [1, 2, 3, 4]

1.1.1 Hardware

Základní hardwarové schéma PLC se skládá z několika komponentů. Konkrétně z procesoru, paměti, napájecí jednotky, rozhraní pro vstupy, výstupy a komunikaci a zařízení pro programování.

- CPU, neboli centrální procesorová jednotka obsahuje mikroprocesor. Tato jednotka zpracovává vstupní signály a zpracovává je podle předepsaného programu v paměti a mění výstupy.
- Napájecí jednotka je potřebná pro převod střídavého napětí elektrické sítě na nízké stejnosměrné napětí pro vnitřní obvody.
- Programovací zařízení slouží k zadání příslušného programu do paměti. Program je vyvíjen externě v zařízení a později je přenesen do paměti procesoru.

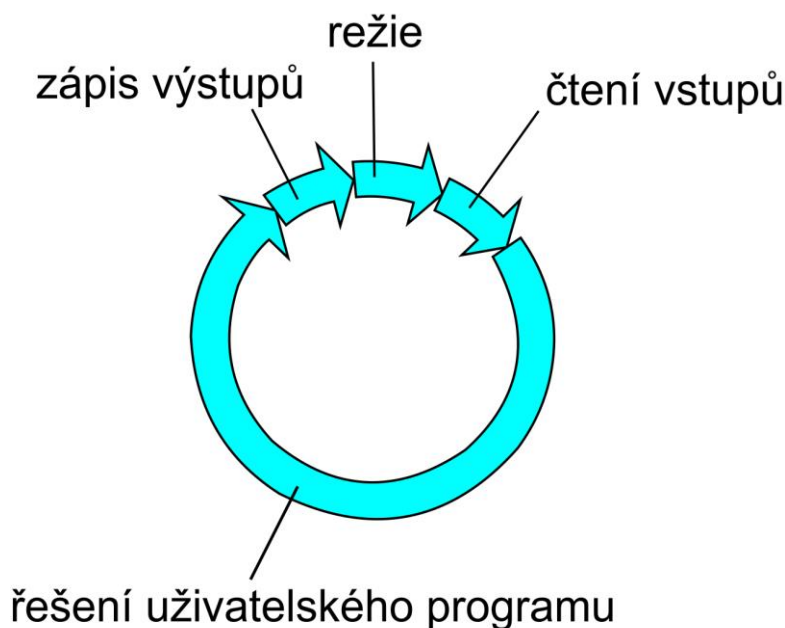
- Paměťová jednotka je místo, kde je uložen program obsahující řízené akce, které mají být vykonány mikroprocesorem. Také jsou zde uloženy data ze vstupů a zapsána data pro výstupy.
- Vstupní a výstupní jednotky jsou oblasti, odkud procesor přijímá informace z externích zařízení a odkud jim je naopak posílá. Zdroj vstupních signálů může přijít například ze senzorů polohy, teploty, tlaku, průtoku nebo jen přepínače, či tlačítka. Výstupní signály z PLC mohou sloužit například jako ovládací prvky pro startovací cívky motoru, či elektromagnetické ventily. Tyto signály mohou být diskrétní, či spojité povahy.
- Komunikační rozhraní slouží k přijímání, či odesílání dat na komunikačních sítích nebo pro komunikaci s dalšími PLC, či jinými zařízeními, jako jsou např. ovládací panely.[2]



Obrázek 1 – Hardwarové schéma PLC [2]

1.1.2 Zpracování programu

Program pro PLC je posloupnost instrukcí a příkazů daného jazyka. Systémový program zaručuje to, že uživatelský program je vykonáván cyklicky, tedy od začátku do konce ve smyčce. U většiny PLC se jedna otočka cyklu skládá ze čtyř částí. V první části se aktuální hodnoty výstupů zapišou do obrazů výstupů, ve druhé se provedou režijní systémové operace, což mohou být například aktualizace systémových a časových proměnných nebo naplánována aktivace procesů pro další cyklus. V další části jsou aktuální hodnoty vstupů zapsány do obrazů vstupů a poté je předáno řízení uživatelskému programu.



Obrázek 2 – Cyklus PLC programu

Důvod používání obrazů vstupů a výstupů je synchronizace vstupních a výstupních dat s běžícím programem, zamezuje se tak vzniku chyb způsobených různě se měnícími dat. [1, 3, 5]

1.1.3 Programování

Programování může probíhat na programovacích zařízeních. To může být příruční zařízení, stolní konzole nebo osobní počítač. Každá firma vyrábějící PLC vyvíjí svůj programovací software. Uživatelský program může být napsán v různých jazycích. O jejich sjednocení a popis se stará mezinárodní norma IEC 1131-3. Přenositelnosti mezi jednotlivými výrobci není zajištěna. Těchto jazyků je pět.

- *Kontaktní schéma (Ladder Diagram) LD* – Jedná se o liniová, neboli kontaktní schémata. Výhoda byla podobnost s reléovými ovládacími obvody, tedy technika se dala lehce naučit. Jedná se o ekvivalenci soustavy logických rovnic. Nevýhoda byla nutnost vyhodnocování všech příkazů, i těch zrovna nepotřebných, což se dá nahradit možností skákat v programu, což může vyvolat fatální chyby. Jedná se o metodu dnes již zastaralou a případně použitelnou v jednodušších úlohách.
- *Funkční bloky (Function Blocks) FB* – Je to obdobný jazyk jako LD, též se jedná o grafickou strukturu. Logické operace znalé z projektování ovládacích schémat z integrovaných obvodů jsou zde, stejně jako čítače, časovače, či aritmetické

operace, znázorněny jako bloky s danou funkcí. Vhodný k programování je také díky své přehlednosti.

- *Seznam příkazů (Instruction List) IL* – Tento jazyk je obdoba jazyka Assembler u počítačů, je tedy strojově orientován. Použití v dnešní době není moc běžné, díky své náročnosti. Použití může být u nejjednodušších systémů nebo tam, kde je kladen důraz na rychlost.
- *Strukturovaný text (Structured Text) ST* – Tato metoda se dá přirovnat k vyšším programovacím jazykům pro PC – Pascal, C. Vůči jazyku IL umožňuje úspornější a přehlednější zápis jednodušších i složitějších algoritmů.
- *Sekvenční funkční graf (Sequential Function Chart) SFC* – Vychází z metody GRAFCET. Také je to grafická metoda. Jedná se o sekvenční programování a podporuje systémový přístup k programování. Slouží k hrubému návrhu algoritmu. K popisu struktury používá značky stavů, přechodů a větví. Jednotlivé stavy jsou potom algoritmicky řešeny jiným jazykem. [1,2,4,5]

1.2 Automatizační prostředky značky Siemens

Firma Siemens v Česku působí od roku 1890. Dnes firma patří mezi největší elektrotechnické firmy v České republice. Oblast uplatnění firmy je velmi široká, působí v instalaci *offshore* větrných elektráren, jako dodavatel paroplynového zdroje a technologie pro přenos energie, dále veřejnou infrastrukturou, průmyslovou automatizací a softwarovým řešením. Zároveň je předním dodavatelem zdravotnických zobrazovacích zařízeních a technologií pro laboratorní diagnostiku.

Firma Siemens disponuje pro automatizační systémy velkou škálou prvků. Jaký zvolit prostředek záleží na konkrétním použití, tedy na potřebném výkonu, šířce daného projektu, cíleném prostředí. [6]

Zapůjčené PLC, na kterém probíhal vývoj programu, byl řady S7-1500F, typ CPU 1516F-3 PN/DP. Proto je tento produkt detailně popsán.

1.2.1 PLC SIMATIC S7-1500F, CPU 1516F-3 PN/DP

Konkrétně se jedná o model 6ES7516-3FN01-0AB0. Jednotka je navržena pro mimořádně náročné aplikace se zabezpečením proti poruchám s označením Fail-safe, pro standartní aplikace a pro dodatečné komunikační úlohy. [7]



Obrázek 3 – CPU 1516F-3 PN/DP

Přední strana

LED diody znázorňují operační mód a diagnostický status CPU. Přepínač pro změnu módu PLC RUN/STOP/MRES. Objevuje se zde i vytištěná MAC adresa, slot na paměťovou kartu SMC, konektor pro napájení, konektor na PROFIBUS, konektory na 2x PROFINET s LED diodami pro kontrolu připojení a utahovací šrouby. Dále je zde konektor pro připojení přední desky s displejem a šesti ovládacími prvky.

Displej

Displej o úhlopříčce 6,1 cm zobrazuje jednoduché menu o 5 položkách:

Overwiev - lze zjišťovat softwarové i hardwarové informace o PLC, Fail-safe programu, výši zabezpečení a stavu paměťové karty

Diagnostics – zde lze najít historii požadavků na PLC, nastavené upozornění, prohlížet takzvaných *Watch tables* a časy cyklů a další informace o paměti

Settings – v této položce lze nastavovat a prohlížet adresy, čas a datum, měnit režimy PLC a zabezpečení, resetovat paměť, uzamykat displej, či aktualizovat firmware

Modules – položka obsahuje informace o připojených modulech

Display – zde lze nastavit vlastnosti displeje a najít výrobní informace daného typu displeje

Napájení, firmware a paměť

Napájení přístroje je pro 24 V DC. Firmware verze produktu byla aktuálně V1.8. Integrovaná paměť obsahuje 1,5 MB pro program a 5 MB pro data, nutné rozšíření paměti je pomocí SIMATIC Memory Card s maximální velikostí 32 GB (při práci byla použita karta o velikosti 12 MB).

Procesor, čítače, časovače, adresy

Procesorový čas pro zpracování operace jednoho bitu je 10 ns. S7 čítače a časovače jsou omezeny každý počtem 2048, IEC časovače a čítače jsou omezeny jen pamětí. Počet IO modulů je omezen na počet 8192. Oblast IO adres je pro vstup i výstup pro každý 32 kB.

Komunikace

DP master a IO kontrolér je integrován v počtu jedna a oba jsou rozšiřitelní komunikačními moduly na počet 8. Počet modulů na jeden nosič „rack“ je omezen na 32. Komunikační rozhraní probíhá pomocí dvou konektorů RJ45 sběrnici PROFINET a konektorem RS 485 sběrnici PROFIBUS pracující jako DP master.

Přídavné moduly

Mezi možné rozšíření moduly patří regulátory PID_Compact, PID_3Step, PID-Temp, tedy PID regulátor univerzální, se zaměřením na ventily, na teplotu.

Pracovní podmínky

Produkt je schopný pracovat v horizontální poloze v rozmezí 0-60°C, ve vertikální poloze 0-40°C.

Programovací jazyky

Programování probíhá v programu TIA portál s podporou jazyků STL, SCL, GRAPH a s podporou fail-safe LAB a FBD. Pro bezpečnost lze nastavit ochranu proti editaci a kopírování programu a ovládání displej. Lze nastavit i úroveň tohoto zabezpečení. [8]

1.2.2 Rozšiřovací moduly pro SIMATIC S7-1500F

Signální moduly

Slouží jako prostředník mezi PLC jednotkou, tedy ovládáním a snímaným procesem. Signální moduly se dělí podle určitých druhů. Digitální vstupy mohou být v počtu 16 nebo 32. Vstupní napětí je buď 24V DC nebo 120/230V AC. Digitální výstupy mohou být

v počtu 2, 16, 32. Výstupní napětí může být 24 V DC nebo 230 V DC a proudy 0,5 A, 2 A, 5 A. Počet analogových vstupů je 8, výstupů pak 4 nebo 8.

Technologické moduly

Tyto moduly nabízejí zpracování signálu na úrovni hardwaru, slouží pro rychlé počítání, měření a sledování polohy pro velkou škálu snímačů. Poloha může být získána z inkrementálních, či absolutních enkodérů.



Obrázek 4 – Přídavné moduly pro S7-1500

Komunikační moduly

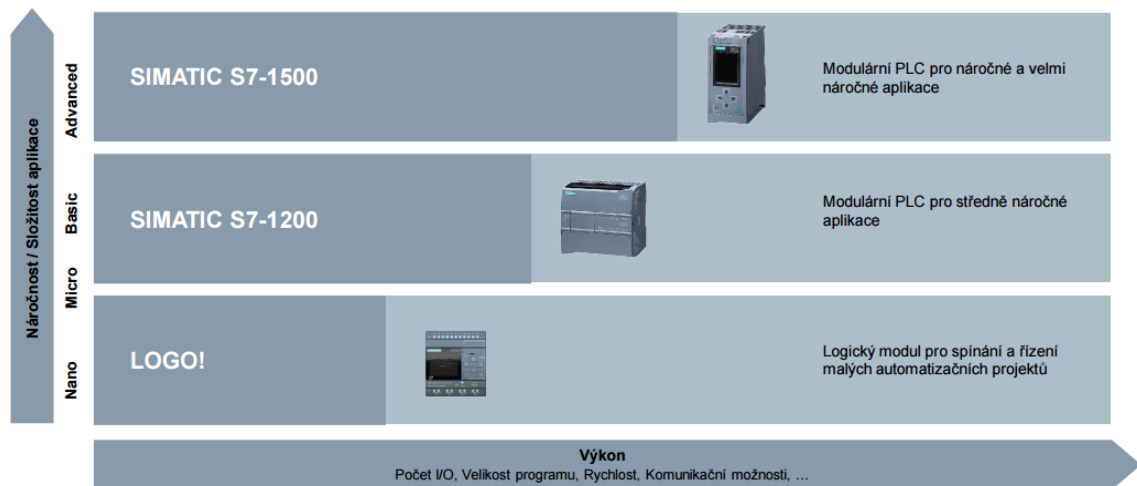
Tyto moduly zvyšují kapacitu komunikačních rozhraní. Rozšířit se může počet konektorů pro použití sběrnice PROFINET nebo PROFIBUS. Další možnost je připojení modulu pro ptp komunikaci. To může sloužit například pro připojení čárových kódů nebo RFID čteček.

HMI panely

Nabízeny jsou KEY panely, jedná se o panely pouze s klávesami, které slouží pro operátora. Další z nabídky jsou BASIC panely sloužící pro vizualizaci a monitorování. Označení COMFORT kombinuje předešlé dva produkty, tedy obrazovku a klávesy. Posledním produktem je mobilní ovládací panel MOBIL a tenký ovládací dotykový panel THIN. [7]

1.2.3 Ostatní produkty firmy Siemens a označení

Firma Siemens disponuje pro automatizační systémy velkou škálou prvků. Jaký zvolit prostředek záleží na konkrétním použití, tedy na potřebném výkonu, šířce daného projektu, cíleném prostředí.



Obrázek 5 – Ostatní PLC zástupci podle kategorií [9]

Firma dělí své produkty na kategorie *Nano*, *Micro*, *Basic* a *Advanced*. Tedy podle poměru pravděpodobného umístění pro různé náročné projekty a pro šířku daného projektu.

Řada S7-1500 je nejnovější PLC této firmy a jak je už bylo zmíněno, hodí se na nejnáročnější úlohy. U řady S7-1200 se jedná o základní řídicí systém pro malé projekty. Základní modul je k dispozici v 5 verzích v závislosti na požadovaném výkonu, je k dispozici i ve formě SIPLUS. Produkty s označením LOGO! jsou určeny pro ještě méně náročnější aplikace.[9]

Označení SIPLUS

Již více zmíněné označení u jednotlivých řad PLC značí zvýšenou ochranu vůči extrémním podmínkám. Je tedy možné provozovat PLC v širším pásmu teplot např. -40°C až $+70^{\circ}\text{C}$. Dále je možnost pracovat v místech se zvýšenou kondenzací a vlhkostí. Je zaručena zvýšená míra ochrany proti vodě a prachu nebo mechanickému zatížení. PLC je schopen také pracovat s nestandardními rozsahy napětí.

Fail-safe

Takto označené PLC jednotky splňují normu IEC 61508 o bezpečnosti elektrických, elektronických a programovatelných elektronických systémů. Jsou také vhodně k použití

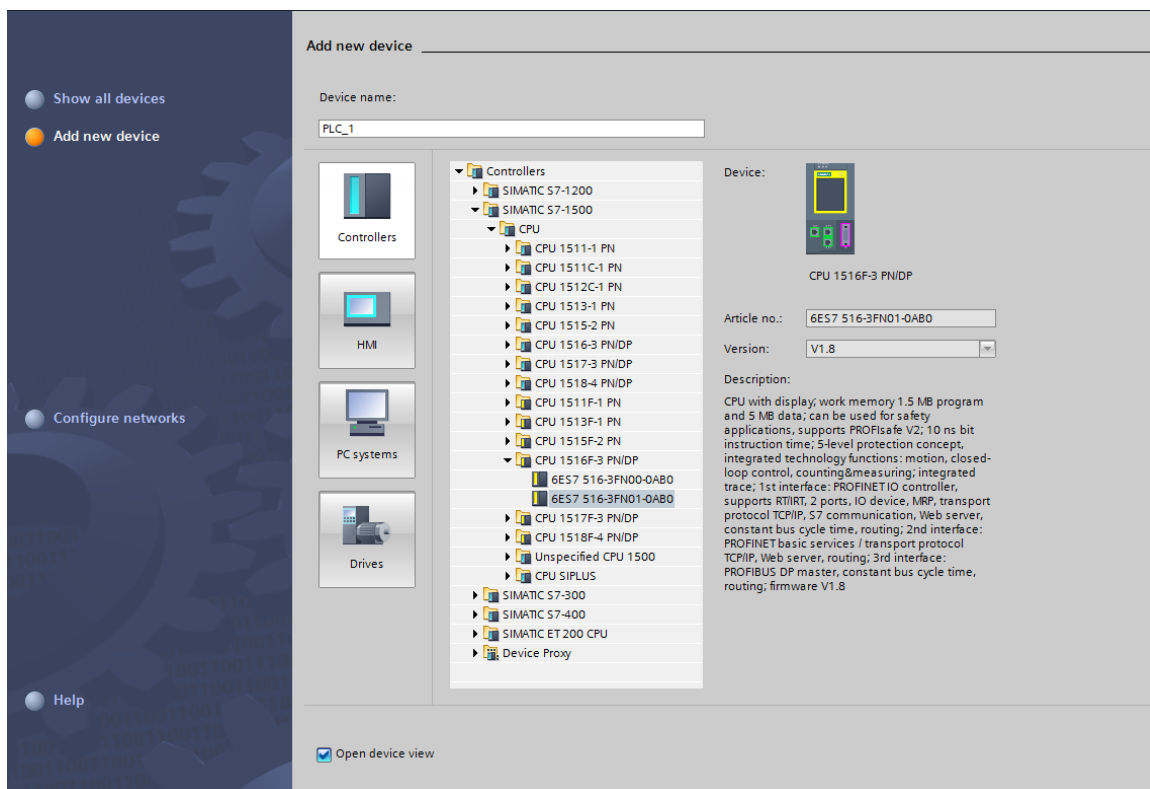
v aplikacích s označením až SIL3 (úroveň integrity bezpečnosti) v souladu s normou IEC 62061 a s označením PLe v souladu s normou ISO 13849, tedy PLC dosahuje nejvyšší úrovně bezpečnosti. [7,10,11]

1.3 TIA Portál

Celým názvem *Totally Integrated Automation Portal* je programovací nástroj pro vytvoření požadovaných úloh. Jedná se o multiplatformní software. Tento software sjednocuje do jednoho prostředí SIMATIC systémy, jako je STEP7 pro programování PLC, WinCC pro programování HMI zařízení a StartDrive sloužící k parametrizaci Siemens pohonů.

1.3.1 Nový projekt

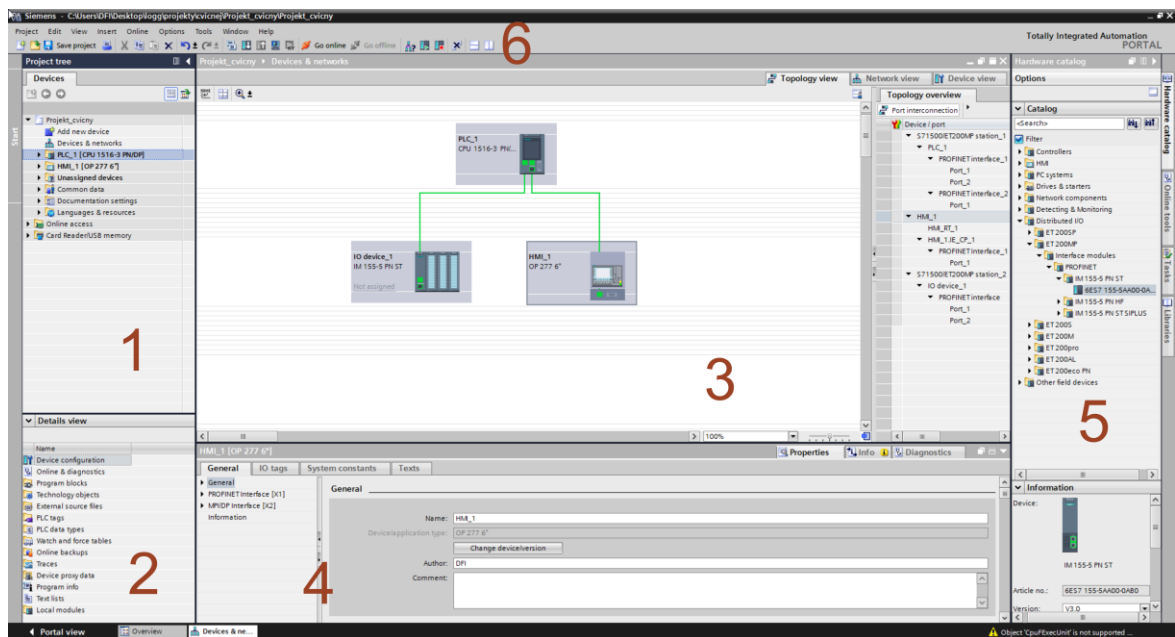
Po zapnutí programu se objeví nabídka o vytvoření, otevření nebo migraci projektu. Migrace slouží k převedení programu ze starších verzí STEP7 do prostředí TIA Portál. Po vytvoření projektu je potřeba zvolit zařízení, které se budou v projektu používat. Na výběr jsou PLC, HMI panely, počítače a pohony.



Obrázek 6 – Výběr zařízení

1.3.2 Pracovní prostředí

Toto prostředí se skládá s několika částí. V oblasti 1 se nachází větvící strom aktuálního projektu, kde jsou jednotlivé vybrané zařízení. Tyto zařízení jsou reprezentovány jako složky a každá má svou podložku s konfiguračními, či programovacími soubory. Oblast 2 zobrazuje podsoubory označené složky.



Obrázek 7 – Náhled pracovního prostředí

Oblast 3 je hlavní pracovní plocha, kde dochází k práci podle aktuální akce, například k programování, náhledu na zařízení, úpravě topologie zařízení. Oblast 4 slouží k zobrazování podrobností a nastavení zařízení, či podrobnosti o stavu kompilování programu. Oblast 5 obsahuje knihovny funkcí při programování nebo katalog produktů u editaci zařízení.



Obrázek 8 – Vizualizace PLC

Poslední oblast 6 je klasické menu, například lze zde pracovat s projektem, měnit rozestavění oken obrazovky, měnit režim PLC, přecházet do stavu online, kompilovat program, nahrávat program do PLC nebo naopak z PLC. Možnost je i spustit simulátor PLC.

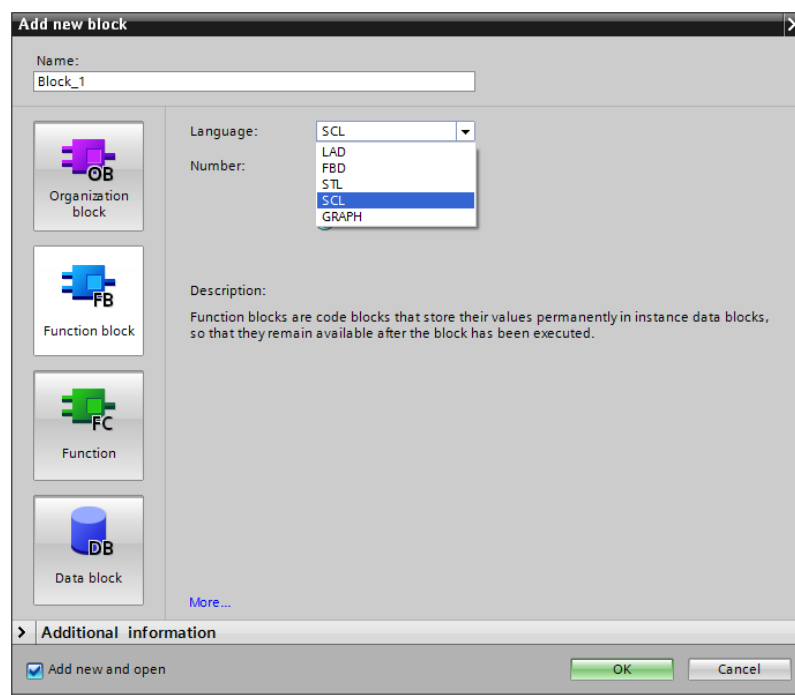
Vybrané PLC je zobrazeno na DIN liště, k němu lze doplnit další moduly. Na jedné liště lze mít pouze jedno PLC. U toho lze nastavit komunikační rozhraní, minimální a maximální čas pro jeden cykl, nastavení webserveru, nastavení displeje, nastavení ochrany přístupu k PLC včetně nastavení hesla. [12]

1.3.3 Tvorba programu

Program se vytváří v podložce daného PLC s názvem *Program blocks*.

Program se může skládat z několika souborů:

- Organizační bloky – systémové předprogramované bloky
- Funkční bloky – uživatelsky programovatelné bloky spjaté s datovým blokem, kde jsou permanentně uložená data
- Funkce – uživatelsky programovatelné bloky s dočasnou pamětí
- Datové bloky – blok s uloženými daty

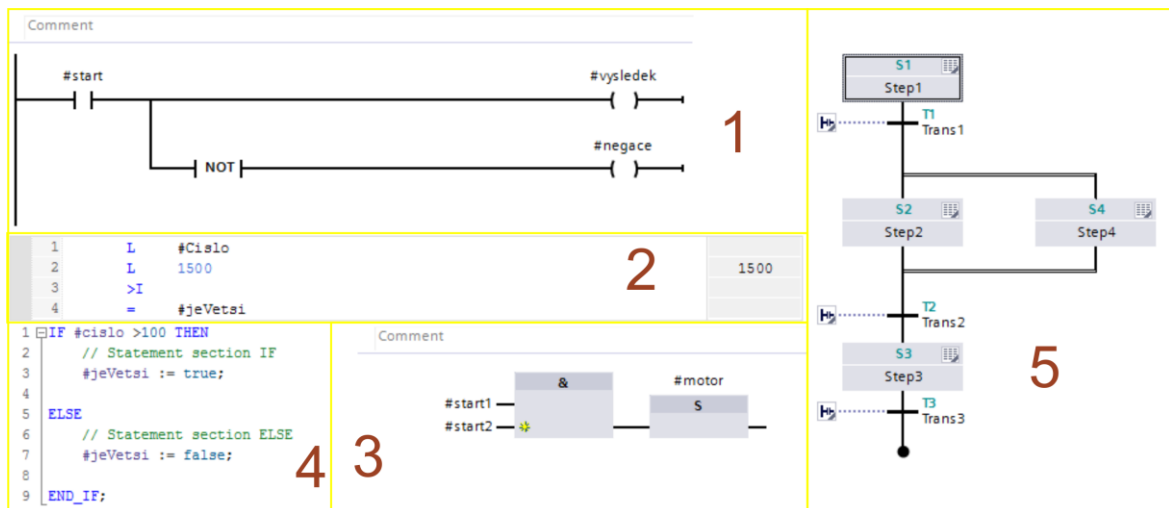


Obrázek 9 – Programové bloky

Jako výchozí blok zde slouží systémový *Main*, odkud se dají spouštět ostatní bloky. K dispozici je pět programovacích jazyků, které odpovídají předepsané normě IEC 1131-3:

1. LAD – Jedná se o jazyk *Kontaktní schéma LD*
2. STL – Jedná se o jazyk *Seznam příkazů IL*
3. FBD – Jedná se o jazyk *Funkční bloky FB*
4. SCL – Jedná se o jazyk *Strukturovaný text ST*
5. GRAPH – Jedná se o jazyk *Sekvenční funkční graf SFC*

Ukázka jazyků je na následujícím obrátku.



Obrázek 10 – Programovací jazyky

1.3.4 Webserver

K PLC lze přistoupit pomocí protokolu PROFINET s pomocí PC, či notebooku přes webové rozhraní. K přístupu je potřeba zadat IP adresu daného PLC a mít aktivován webový server a povolení na PROFINET rozhraní.

V PLC lze nastavit uživatelské účty pro zabezpečení přístupu. Po přihlášení web nabízí přehled o stavu PLC, diagnostiku, přehled o využití paměti, přehled modulů, komunikační statistiku. Dále umožňuje číst a zapisovat do proměnných a procházet uživatelsky definovanou tabulku s proměnnými – *Watch tables*. Další možnost je procházet adresář souborů paměťové karty a uživatelsky vytvořené CSV soubory.

Důležitá funkce v této práci využívána je možnost vytvoření uživatelské stránky. Uživatelem vytvořené zdrojové soubory musí být načtené do PLC, kde se z nich stanou

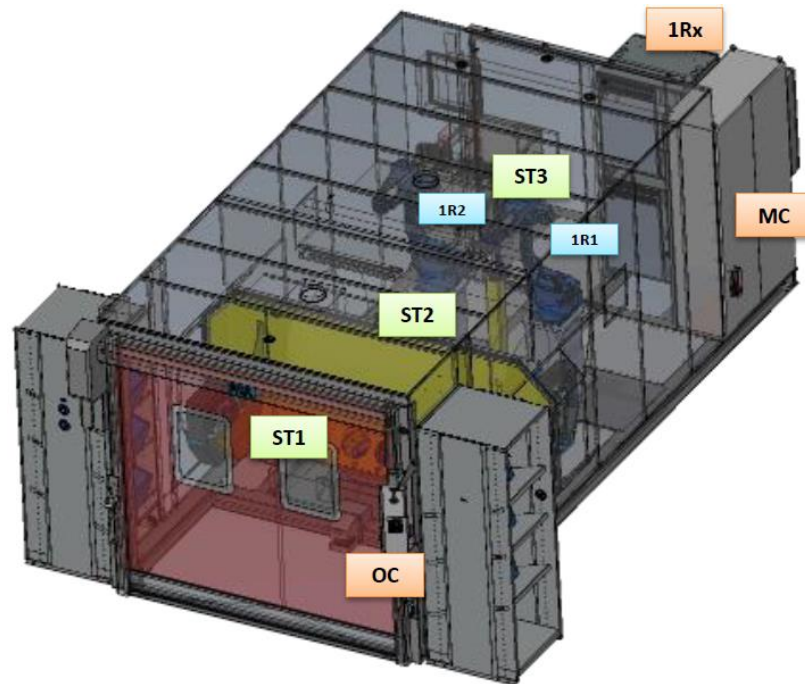
datové bloky a jsou uloženy v paměti. Pak lze k této stránce přistupovat pomocí vestavěného webserveru.

The screenshot displays the Siemens SIMATIC S7-1500 webserver interface. At the top, the Siemens logo and station name 'S71500/ET200MP station_1/PLC_1' are visible. The interface includes a user login section on the left with a 'User name' field and a 'Log in' button. A navigation menu on the left lists various functions: Start page, Diagnostics, Diagnostic Buffer, Module information, Alarms, Communication, Topology, Tag status, Watch tables, Customer pages, Filebrowser, DataLogs, and Introduction. The central area features a virtual CPU operator panel with a display showing 'SIMATIC S7-1500', 'RUN' mode, 'CPU 1516F-3 PN/DP', and the hardware ID '6ES7 516-3FN01-0AB0'. Below the display are directional arrow keys, an 'ESC' key, and an 'OK' key. On the right side, configuration and status information is provided. The 'General' section includes TIA Portal (V13.0 SP1), Step 7 Safety (V13.0 SP1), Station name (S71500/ET200MP station_1), Module name (PLC_1), and Module type (CPU 1516F-3 PN/DP). The 'Status' section shows Operating Mode (RUN), Status (OK), and Mode selector (RUN). The 'Fail-safe' section includes Safety mode, Collective F-signature, and Last fail-safe modification. At the bottom right, a 'CPU operator panel' section contains buttons for 'RUN', 'STOP', and 'LED flashes'.

Obrázek 11 – Náhled na webserver

2 POPIS SVAŘOVACÍ BUŇKY

Svařovací buňka je výrobek firmy AWL-Techniek. Jedná se o standardizovanou buňku s označením PN23345.



Obrázek 12 – Svařovací buňka PN23345

Tato buňka je rozdělena na osm částí podle svých funkcí.

Tabulka 1 – Sekce svařovací buňky

Sekce	Název	Význam
1	OC	Operátorský panel (HMI, ovládací tlačítka)
2	ST1	Jig stanice v prostoru operátora
3	ST2	Jig stanice v prostoru robota
4	ST3	Svařovací stanice
5	1R1	Robot 1
6	1R2	Robot 2
7	MC	Hlavní rozvaděč
8	1Rx	Rozvaděč robotů

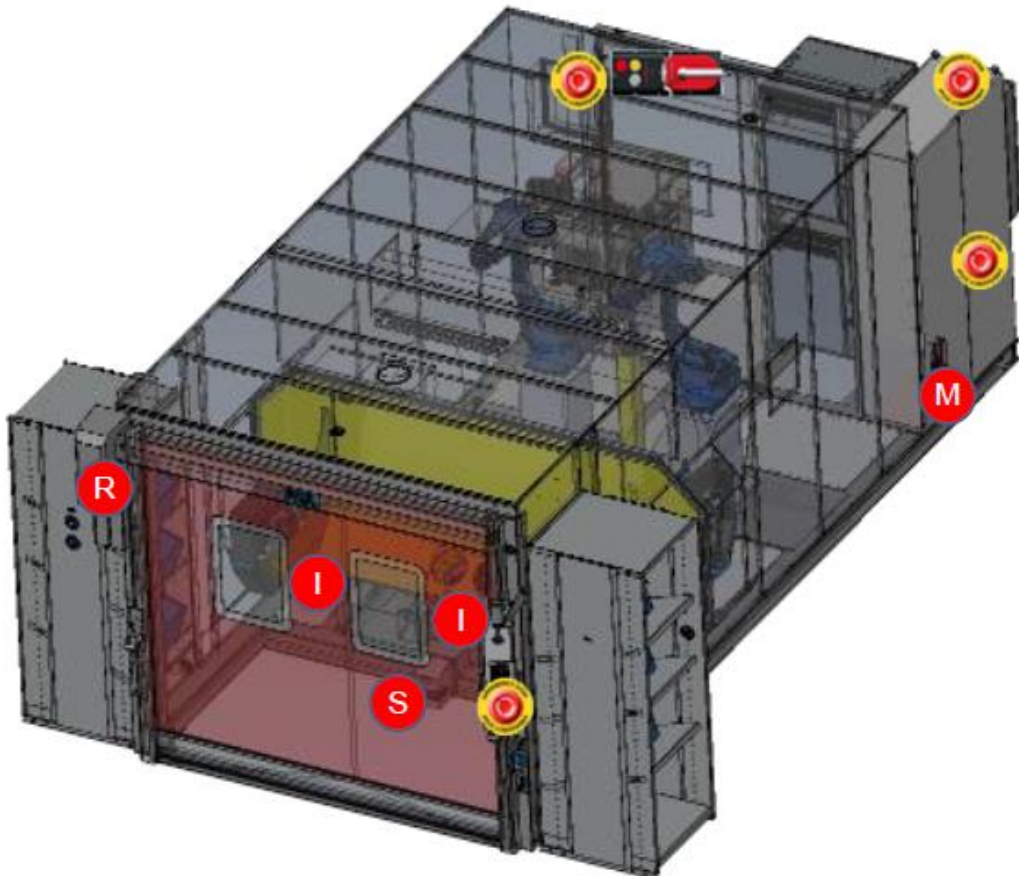
U této buňky je jeden otočný stůl s dvěma jiggy na každé straně. Jedná se upínací stoly pro

přípravky, dále jen jig. Dva jsou z důvodu, aby se mohl jeden nakládat, zatímco se druhý svařuje. U nestandardních buněk může mít jeden otočný stůl například 4 stanice, tedy 4 jigy, kde každá má svého operátora. Tato buňka používá obloukové svařování, ale může být i pro laserové a bodové.

2.1 Vnější ovládací prvky







Svařovací buňka disponuje několika ovládacími prvky. Některé jsou bezpečnostního charakteru, některé pro rozhraní člověk-stroj, dále vedeno jako HMI (*Human Machine Interface*).

2.1.1 Bezpečnostní prvky



Obrázek 13 – Svařovací buňka s bezpečnostními prvky

Svařovací buňka má několik bezpečnostních prvků, které umožňují v případě nouze vypnout systém, zabraňují přístupu dovnitř cely nebo hlídají nebezpečný prostor v oblasti jig stanice v prostoru operátora a zabraňují tak úrazu.

-  Hlavní přepínač
-  Nouzové stop tlačítka
-  Uzavření servisních dveří
-  Rolovací dveře
-  Skener bezpečné oblasti
-  Bezpečnostní senzory polohy otočného stolu

Další zabezpečení se týká uživatelského přístupu a hesel ke změnám nastavení buňky. Některá nastavení v HMI mohou být změněna pouze s určitým oprávněním. Každý bezpečnostní level má své heslo.

Tabulka 2 – Povolení přístupu

Bezpečnostní level	Povolení
awl	Jen pro zaměstnance AWL. Je umožněn přístup ke všem nastavením.
administrator	Určeno pro důležité změny. Určeno jen pro důvěrný a prověřený personál. Bezpečnostní level je stejný jako pro awl, s výjimkou změny hesla pro level awl.
maintenance	Určen pro personál údržby.
operator	Tento level je určen pro operátora pro provádění potřebných nastavení. Heslo slouží také aby nedošlo k náhodným změnám.

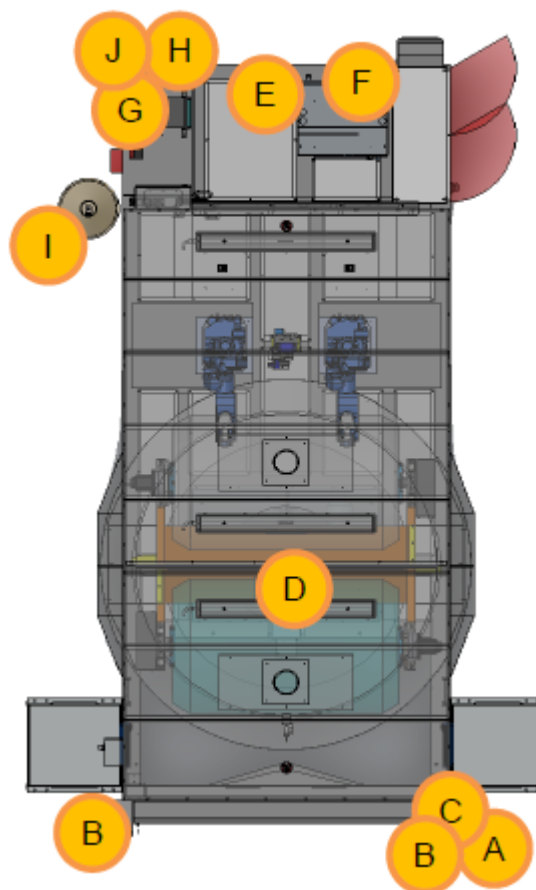
2.1.2 Human Machine Interface

Svařovací stanice disponuje několika HMI prvky, což jsou uživatelsky dostupné ovladatelné prvky různých charakterů.

A – Jedná se o operátorský panel, který obsahuje ovládací dotykový panel firmy Siemens TP900 9“, dále ovládací tlačítka: Nouzové stop tlačítko, Přepínání automatického a manuálního režimu, Tlačítko pro servis, Startování automatického režimu, Tlačítko pro zastavení a restartování, Tlačítko pro resetování zpráv, či chyb, Tlačítka pro ovládání učení.

B – Spínač pro spuštění výrobního cyklu. Tyto spínače jsou po obou bocích operátorského prostoru. Mají stejnou funkci.

C – Signalizační sloupek slouží k indikování globálních statusů stroje pro diagnostiku na delší vzdálenost. Sloupek má tři barvy – modrá, žlutá, červená. Každá barva může svítit nebo blikat s frekvencí 0,5 Hz.



Obrázek 14 – Svařovací buňka s HMI prvky

- D** – Nakládací asistenční světlo. Pomáhá operátorovi signalizovat, co dělat, například - nakládání, vykládání, čekání na start cyklu. Barvy jsou zelená, žlutá, červená. Svítící nebo blikající s frekvencí 1 Hz.
- E** – Tento prvek je bezpečnostní dveřní systém. Servisní dveře svařovací buňky jsou zavřeny a zamčeny pomocí AWL standardizované kliky se zámkem značky Euchner, typ MGB-L2HE-AR-R. Klika má masivní zamykací pin a RFID bezpečnostní spínač nezávislý na zamykacím pinu. Obsahuje Nouzové stop tlačítko a svítící tlačítka pro odemykání a ukazující stav buňky a dveří.
- F** – Programovací jednotka (*teach pendant*) slouží mimo jiné pro případ manuální ovládní robotů. Například při restartu stroje se robot vrací do výchozí polohy. Pokud se nachází v neznámé poloze, musí se vrátit do výchozí polohy manuálně. Programovací jednotka také může sloužit pro servisní účely.



Obrázek 15 – Bezpečnostní dveřní systém

G – Řídicí systém s dotykovou obrazovkou pro konfiguraci a vizualizaci svařovacího procesu.

H – Zde je sklad svařovacího plynu a obsahuje snímače průtoku, tlaku a plynový ventil řízený servomotorem.

I – Zde senzor indikuje stav svařovacího drátu v zásobníku.

J – Servisní jednotka zásobuje čistý stlačený vzduch dovnitř buňky. Zároveň snímá tlak v pneumatickém okruhu.

2.2 Vybavení buňky

Svařovací buňka je složena s dílčích zařízení, které mezi sebou komunikují. Hlavním řídicím prvkem je PLC firmy Siemens.

2.2.1 Seznam zařízení

Tabulka 3 – Název a popis zařízení

Název	Popis
Siemens S7-1517F PN/DP	Bezpečnostní PLC řídící průběh celé buňky
Siemens ET200sp 2x	Decentrální periferie s I/O, komunikačními a technologickými moduly.
Siemens TP900 9"	Operátorský panel s dotykovou obrazovkou a ovládacími tlačítky pro chod cely.
Signal Column	Signalizační sloupek pro signalizaci statusu chodu cely.
Loading Column	Nakládací asistenční světlo pro operátora, udávající co dělat.
Cycle Start L	Spínač pro spuštění výrobního cyklu umístěn na levé straně operátorského prostoru.

Název	Popis
Cycle Start R	Spínač pro spuštění výrobního cyklu umístěn na pravé straně operátorského prostoru.
Siemens Scalance X-300	Pro Profinet/Ethernet propojení prvků cely.
ABB IRB1600ID 2x	Obloukový svařovací robot.
ABB MU200 2x	Standardizovaná motorová jednotka otáčející se stolem, sloužící jako externí osa robota.
ABB IRC5	Řídicí systém robota.
SKS FB5-PN 2x	Průmyslová sběrnice pro komunikaci mezi řídicím systémem robota a napájecím zdrojem svařecího systému.
SKS LSQ5	Napájecí zdroje svařovacího systému. Hořák dle potřeby svařování.
SKS PF5	Podavač svařovacího drátu s integrovaným senzorem průtoku plynu.
SKS Q84r + 2x Q81	Centralizovaný řídicí systém s dotykovou obrazovkou a dvěma Q81 svařovacími kartami (pro každý hořák jedna) pro konfiguraci a vizualizaci svařovacího procesu.
AWL Gas system	Standardizovaný systém rozvodu plynu.
IFM I7R201 3x	Induktivní senzor. Kontrola zásoby drátu v zásobníku 2x. Pro určení středového bodu drátu na hořáku.
Thielmann BRG2000G & DA2000E	Čistící stanice a řezačka drátu pro hořák.
Festo CPX	Modulární elektrický terminál. Pracuje s pneumatickými, elektrickými signály s podporou protokolů průmyslových sítí.
Destaco pneumatic power clamps	Pneumatické svěráky pro fixování svařovacích produktů.
Festo KD4-3, KS3-3	Pneumatické zásuvky a konektory rychlospojek.
Festo MSB4	Jednotka pro úpravu stlačeného vzduchu.
M12 M8 Sensors	Induktivní senzor polohy svařovacích produktů.
Siemens Sirius 3RM1307-2AA04	Startér motoru otočného stolu
Expert EDX960	Rotační motor pro otáčení stolu.
Euchner MGB-L2HE-AR-R	Bezpečnostní dveřní systém.
EM-stop	Tlačítka pro nouzové zastavení cely.
IFM GG712S	Induktivní bezpečnostní senzor polohy otočného stolu.
Sick S3000	Bezpečnostní laserový skener. Skenuje prostor

Název	Popis
	operátora.
Albany MCC	Standartní rolovací dveře.
Norton AXR-SV	Osvětlení buňky.
Rittal airco SK3304500	Nástěnná chladicí jednotka hlavního rozvaděče.
Harting Han	Modulární konektor pro jednoduché, rychlé, centralizované propojení otočného stolu s buňkou. Podporuje elektrické signály i průmyslové komunikace.

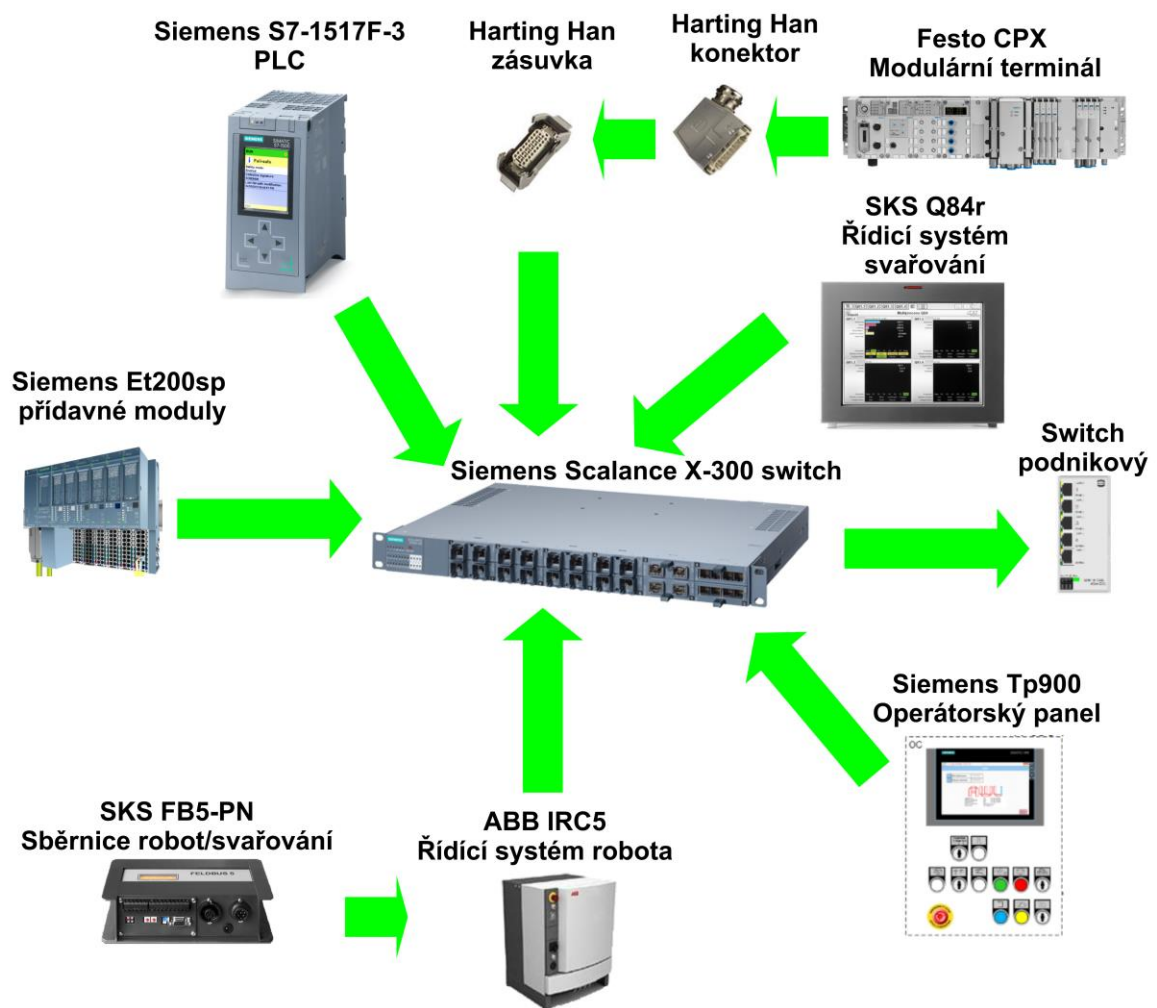
Výše v tabulce uvedené vybavení buňky se může lišit podle přání zákazníka, proto některá zařízení nejsou blíže specifikována. Roboti nemusí být vždy od firmy ABB, ale také od firmy Motoman, od čehož se pak odvíjí změna příslušenství.

2.2.2 Propojení

Většina řídicích prvků je propojena pomocí průmyslové komunikační sběrnice PROFINET. Sběrnice podporuje standart TCP/IP, umožňuje komunikaci v reálném čase, vzdálenou diagnostiku a údržbu, zabezpečení přístupu, vysoký komunikační výkon. Výhoda je také úspora kvůli vedení pouze jednoho kabelu. [13]

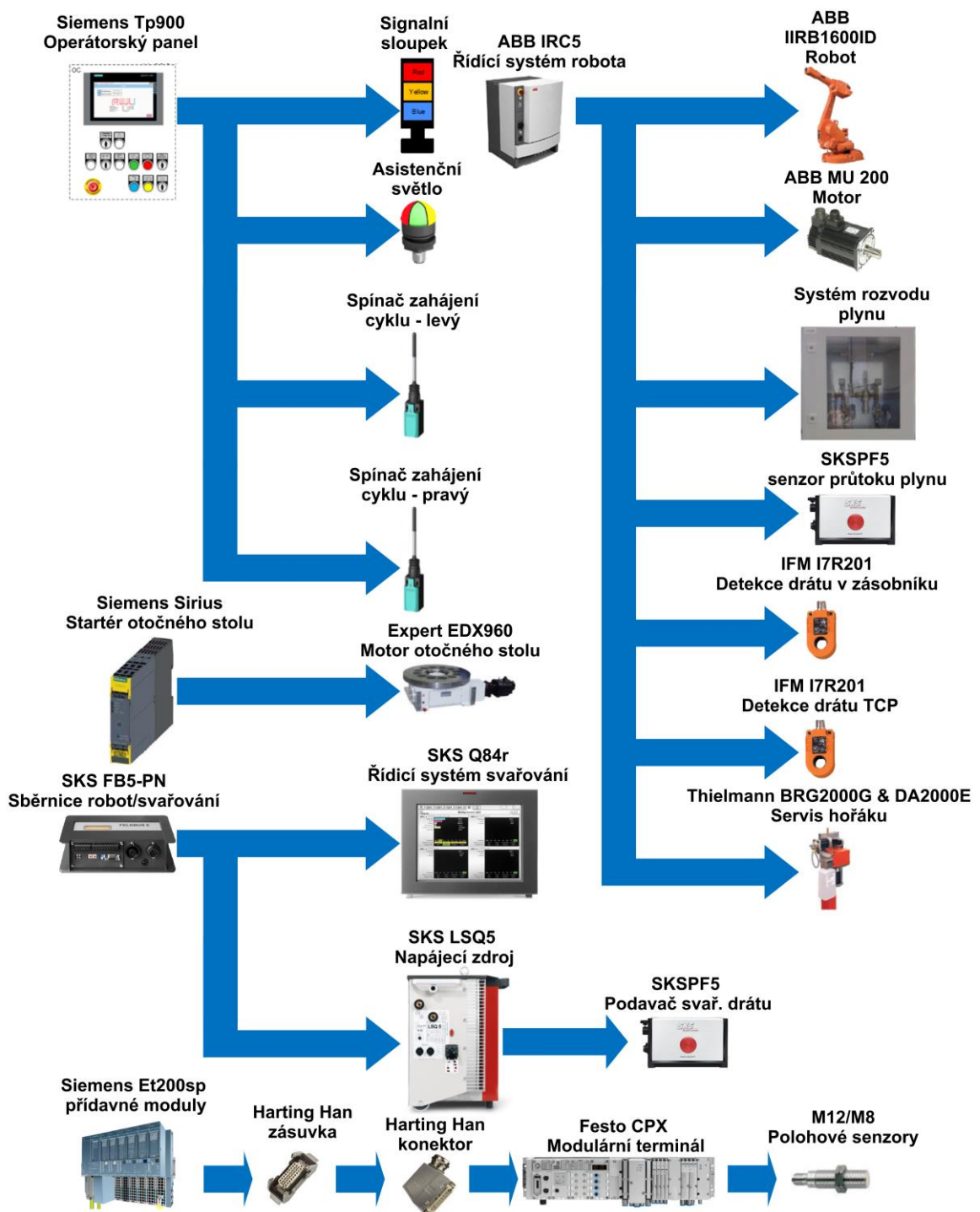
Proto v operátorském kabinetu je umístěn jeden Siemens Et200sp, do kterého vedou kabely z tlačítek a operátorského panelu. Do hlavního rozvaděče vede pouze jeden PROFINET kabel.

Prvky umístěné na svařovacím stole – senzory, pneumatické svěráky jsou pomocí signálních kabelů a pneumatických rozvodů zapojeny do terminálu Festo. Z tohoto terminálu jsou veškeré kabely a komunikace vedeny přes jeden svazek pomocí Harting konektorů do switche. Jak je vidět na následujícím obrázku i ostatní řídicí prvky jsou propojeny do jednoho switche.



Obrázek 16 – Propojení pomocí PROFINET

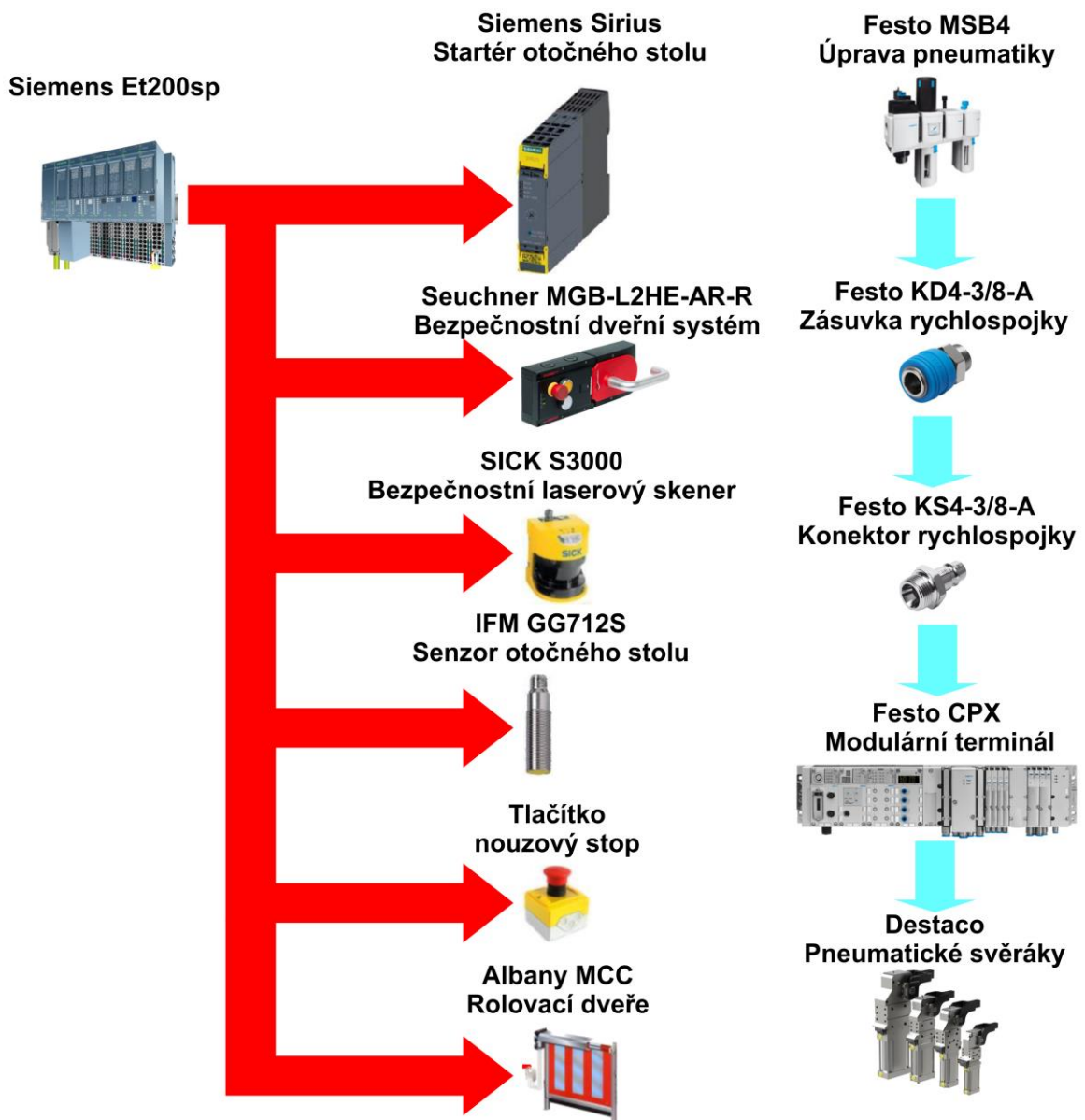
Propojení prvků pomocí kabelů pro napětí 24V je rozděleno na oblasti podle funkčnosti. Operátorský panel je propojen se spínači pro začátek cyklu, signálním sloupkem, asistenčním světlem. Řídicí systém robota je propojen s robotem, motorem pro naklonění stolu, rozvodem plynu, senzorem průtoku plynu, prvkem pro detekci drátu, servisní stanicí robota. Řídicí systém svařování je propojen se sběrnicí pro komunikaci s řídicím systémem robotů a se zdrojem napájení a podavačem svařovacího drátu. Terminále Festo je propojen se senzory polohy svařovacích produktů a pomocí Harting konektorů s přídatnými moduly. Startér je tímto způsobem propojen s otočným stolem.



Obrázek 17 – Propojení pomocí 24V vodičů

Bezpečnostní propojení prvky mají speciální propojení. Znázorněny jsou červenou barvou a místo vedení jednoho signálního kabelu jsou vedeny dva pro menší riziko chyby. Ale jinak se jedná o stejné vodiče, jako v předešlém případě.

Pneumatické prvky jsou propojeny od úpravny pneumatického vzduchu přes rychlospojky do terminálu, který je propojen s pneumatickými svěráky.



Obrázek 18 – Propojení pomocí 24V bezpečnostních kabelů a pneumatiky

Uvedené obrázky propojení jsou schématické a nesignalizují jednoznačně směr komunikace, či směr dat. Některé prvky mohou posílat data v obou směrech. Na otočném stole se nachází dva jigy, ale ve schématech je znázorněno pouze vybavení jednoho jigu, protože jsou totožné. Totéž platí u robota, kde je popsán jen jeden.

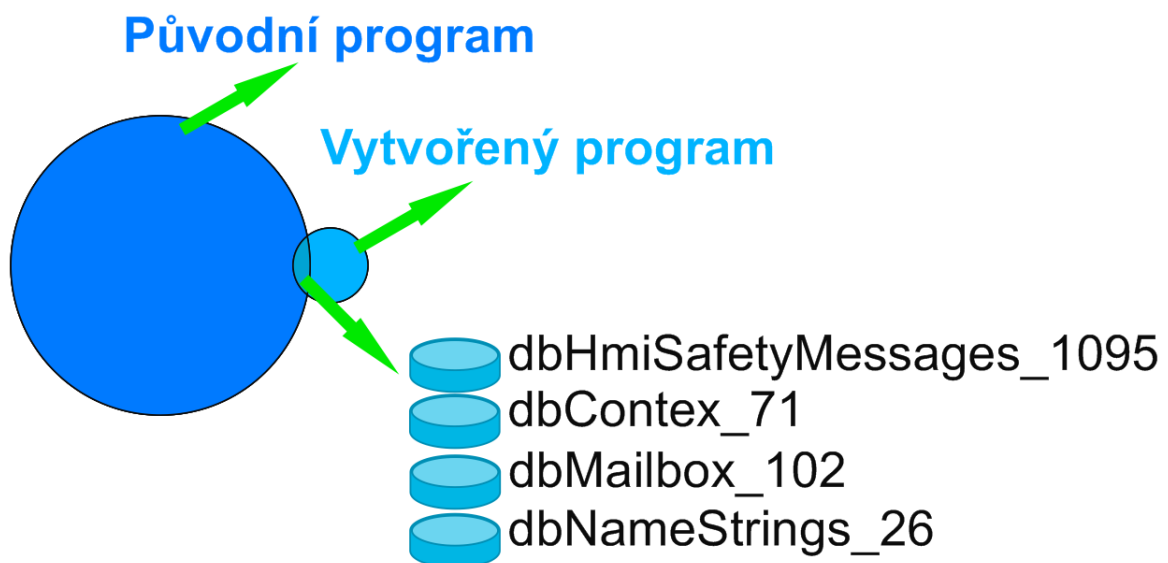
II. PRAKTICKÁ ČÁST

3 POPIS PROGRAMU

V PLC jsou dva programy. Jeden hlavní, který řídí celý proces a bezpečnostní program. V průběhu práce bylo pracováno pouze s hlavním programem. Tyto programy se střídají a bezpečnostní kontroluje po určitém časovém úseku, zda je všechno v pořádku. Pokud dojde k chybě, bezpečnostní program aktivuje příslušnou *SafetyMessage*, což je bezpečnostní událost, která se má monitorovat.

3.1 Původní program

Původní program je velice rozsáhlý a plná orientace v něm by zabrala přibližně rok. Proto bylo zapotřebí se soustředit na nezbytné věci pro splnění zadané práce. Zároveň bylo důležité pokusit se co nejméně zasahovat do původního programu a pokud možno jen číst potřebná data.



Obrázek 19 - Ukázka prolínání programů

Z původního programu jsou využívány jen 4 bloky pro monitorování a čtení dat. Vytvořený program ten tedy původní funkčně rozšiřuje, přičemž dodržuje zásady programování firmy, která využívá způsob psaní proměnných pomocí metody *CamelCase*, tedy psaní víceslovných názvů proměnných bez mezer s velkým počátečním písmenem. Zároveň proměnné musí začínat písmenem identifikující datový typ proměnné (*Integer-i*, *Bool-b*, ...).

3.1.1 dbHmiSafetyMessages_1095

V tomto datovém bloku se nachází bezpečnostní události, které má vytvořený program monitorovat. Všechny proměnné jsou datového typu *Bool*. Celkem je rezervováno místo pro 112 proměnných, z nichž je obsazeno 50.

dbHmiSafetyMessages_1095			
	Name	Data type	Offset
1	Static		
2	Messages	Struct	0.0
3	bMsgButtonEmStopPressedRB1	Bool	0.0
4	bMsgButtonEmStopPressedRB2	Bool	0.1
5	bMsgButtonEmStopPressedRB3	Bool	0.2
6	bMsgFeedbackMC_Indextable_CCW	Bool	0.3
7	bMsgFeedbackMC_Indextable_CCW_CHA	Bool	0.4
8	bMsgFeedbackMC_Indextable_CW	Bool	0.5
9	bMsgFeedbackMC_Indextable_CW_CHA	Bool	0.6

Obrázek 20 – Datový blok HMI Safety Messages

Události mohou být způsobeny zmáčknutím jednoho z nouzových tlačítek. Většina je způsobena tím, že zpětná vazba neodpovídá očekávané hodnotě. Například když je vydán příkaz pro uzavření dveří, ale senzor snímající dveře hlásí, že jsou otevřeny. Takhle jsou kontrolovány hýbající se části – otočný stůl, externí osy robotů. Dále je chyba shody, kdy je zpětná vazba vedena dvěma vodiči a na obou musí být stejné hodnoty. Toto je použito pro snímání operátorského prostoru při otáčení stolu, aby se tam nenacházela žádná osoba. Další typ události je potvrzovací. Při vzniku závady je událost zobrazena na operátorském panelu po dobu trvání. Pokud závada zmizí, objeví se na panelu potvrzovací událost s typem předešlé závady, kterou musí operátor potvrdit pro její zmizení.

Potvrzovací události mají nejnižší prioritu – 1, prioritu 2 mají nouzová stop tlačítka a přepínače. Nejvyšší prioritu mají hýbající se objekty – otočný stůl, dveře, roboti.

Tabulka 4 – Přehled událostí

Událost	Popis	Priorita
bMsgButtonEmStopPressedRB1	Zmáčknuto nouzové Stop tlačítko na řídicím systému Robotu 1	2
bMsgButtonEmStopPressedRB2	Zmáčknuto nouzové Stop tlačítko na řídicím systému Robotu 2	2
bMsgButtonEmStopPressedRB3	Zmáčknuto nouzové Stop tlačítko na řídicím systému Robotu 3	2

Událost	Popis	Priorita
bMsgFeedbackMC_Indextable_CCW	Zpětná vazba otočného stolu proti směru hodinových ručiček	3
bMsgFeedbackMC_Indextable_CCW_CHA	Redundantní zpětná vazba otočného stolu protisměru hodinových ručiček	3
bMsgFeedbackMC_Indextable_CW	Zpětná vazba otočného stolu po směru hodinových ručiček	3
bMsgFeedbackMC_Indextable_CW_CHA	Redundantní zpětná vazba otočného stolu po směru hodinových ručiček	3
bMsgFeedbackOperatorPos1RollupBlockClose	Zpětná vazba zavírání rolovacích dveří.	3
bMsgProfisafeReintegrationNecessary	Nutné znovu načtení bezpečnostního programu.	3
bMsgOperatorPos1HorizontalFloorAreaErrorDiscFault	Chyba nesouladu horizontálního snímání operátorského prostoru.	3
bMsgOperatorPos1HorizontalFloorAreaErrorAckReq	Potvrzení - Chyba nesouladu horizontálního snímání operátorského prostoru.	1
bMsgOperatorPos1VerticalLightScreenErrorDiscFault	Chyba nesouladu vertikálního snímání operátorského prostoru.	3
bMsgOperatorPos1VerticalLightScreenErrorAckReq	Potvrzení - Chyba nesouladu vertikálního snímání operátorského prostoru.	1
bMsgOperatorPos1LockupProtection	Rolovací dveře jsou zavřeny, ale není aktivní skener nebo světelné závory.	3
bMsgButtonEMStopPressedOC1	Zmáčknuto nouzové Stop tlačítko na operátorském kabinetu.	2
bMsgButtonEMStopPressedMC	Zmáčknuto nouzové Stop tlačítko na hlavním rozvaděči.	2
bMsgAckFeedbackTooling3	Potvrzení - Zpětná vazba relé ovládající vzduch jig 3.	1
bMsgFeedbackTooling4	Zpětná vazba relé ovládající vzduch jig 4.	3
bMsgAckFeedbackTooling4	Potvrzení - Zpětná vazba náradí 4.	1
bMsgFeedbackRobotGeneralStop	Zpětná vazba Hlavní vypínač robota.	2
bMsgAckFeedbackRobotGeneralStop	Potvrzení - Zpětná vazba Hlavní vypínač robota.	1
bMsgFeedbackRobotAutoStop	Zpětná vazba Automatický vypínač robota.	2
bMsgAckFeedbackRobotAutoStop	Potvrzení - Zpětná vazba Automatický vypínač robota.	1
bMsgFeedbackExternalAxis1	Zpětná vazba externí osa robota 1.	3
bMsgAckFeedbackOperatorPos1RollupBlockClose	Potvrzení - Zpětná vazba zavírání rolovacích dveří stanice 1.	1

Událost	Popis	Priorita
bMsgFeedbackOperatorPos1RollupDoorEMS	Zpětná vazba rolovací dveře nouzové vypnutí.	3
bMsgAckFeedbackOperatorPos1RollupDoorEMS	Potvrzení - Zpětná vazba rolovací dveře nouzové vypnutí.	1
bMsgFeedbackTooling1	Zpětná vazba relé ovládající vzduch jig 1.	3
bMsgAckFeedbackTooling1	Potvrzení - Zpětná vazba relé ovládající vzduch jig 1.	1
bMsgFeedbackTooling2	Zpětná vazba relé ovládající vzduch jig 2.	3
bMsgAckFeedbackTooling2	Potvrzení - Zpětná vazba relé ovládající vzduch jig 2.	1
bMsgFeedbackTooling3	Zpětná vazba relé ovládající vzduch jig 3.	3
bMsgAckFeedbackIndexTable1Inhibit	Potvrzení - Zpětná vazba napájení otočného stolu.	1
bMsgFeedbackMCEmergencyStop	Zpětná vazba hlavní rozvaděč nouzové vypnutí.	2
bMsgAckFeedbackMCEmergencyStop	Potvrzení - Zpětná vazba hlavní rozvaděč nouzové vypnutí.	1
bMsgFeedbackMCPowerOnTipDresser	Zpětná vazba vyměňování elektrod u bodového svařování.	3
bMsgAckFeedbackMCPowerOnTipDresser	Potvrzení - Zpětná vazba vyměňování elektrod u bodového svařování.	1
bMsgFeedbackMC_MC_MGB_Door_Closed	Zpětná vazba zámku servisních dveří.	3
bMsgFeedbackOC1_Button_Ems	Zpětná vazba tlačítko nouzové vypnutí.	2
bMsgFeedbackOC1Key_Switch_Machine_Mode_Service	Zpětná vazba klíčový přepínač servisu.	2
bMsgAckFeedbackExternalAxis1	Potvrzení - Zpětná vazba externí osa robota 1.	1
bMsgFeedbackExternalAxis2	Zpětná vazba externí osa robota 2.	3
bMsgAckFeedbackExternalAxis2	Potvrzení - Zpětná vazba externí osa robota 2.	1
bMsgFeedbackExternalAxis3	Zpětná vazba externí osa robota 3.	3
bMsgAckFeedbackExternalAxis3	Potvrzení - Zpětná vazba externí osa robota 3.	1
bMsgFeedbackExternalAxis4	Zpětná vazba externí osa robota 4.	3
bMsgAckFeedbackExternalAxis4	Potvrzení - Zpětná vazba externí osa robota 4.	1
bMsgFeedbackIndexTable1Inhibit	Zpětná vazba napájení otočného stolu.	3
bMsgFeedbackOC1Key_Switch_Machine_Mode_Teach	Zpětná vazba klíčový přepínač učení.	2

Událost	Popis	Priorita
bMsgFeedbackOC1Roll_up_Door_Safe_Closed_CHA	Redundantní zpětná vazba rolovací dveře bezpečně zavřeno.	3

3.1.2 dbContext_71

Jedná se o první blok, z kterého jsou čerpána data o stavu buňky. Tento datový blok obsahuje strukturu s proměnnými dle oblasti výskytu. Struktura je zobrazena na následujícím obrázku.

dbContext_71				
	Name	Data type	Offset	Start value
1	Static			
2	AllData	*udtDbContext_72*	0.0	
3	MachineStatus	*udtMachineStatus...	0.0	
4	ObjectStatus	*udtObjectStatus_2...	4.0	
5	iStationNumber	Int	8.0	0
6	iLayoutNumber	Int	10.0	1
7	bOperatorStationLoad	Bool	12.0	TRUE
8	bOperatorStationUnload	Bool	12.1	TRUE
9	bLoadingContinuesAtNextStation	Bool	12.2	false
10	bTeachActive	Bool	12.3	false

Obrázek 21 – Datový blok Context

Podskupina *MachineStatus* obsahuje proměnné popisující stav celé buňky, například je-li buňka zastavena, spuštěna, restartována, je-li v pohotovostním stavu, stav servisních dveří. Stav potvrzovacího tlačítka, jestli mělo poslední skenování bezpečnostního programu chybu, či událost. Zdali bylo stisknuto nouzové stop tlačítko a další. Těchto proměnných je 27.

Podskupina *ObjectStatus* obsahuje proměnné, kam přistupuje některý z objektů cely a například požaduje restart, indikuje, že je v režimu učení nebo došlo k chybě v objektu, či potřebuje potvrzení zprávy na HMI zařízení. Počet proměnných je 10.

Zbylé proměnné datového bloku jsou obecné, zaznamenávají se pouze, či dochází k nakládání, vykládání, či nakládání pokračuje na další stanici. Tedy 3 proměnné.

3.1.3 dbMailbox_102

Druhý datový blok, odkud se čtou data o stavu buňky. Nachází se zde 30 schránek. Každá schránka patří napořád jednomu objektu. V této buňce je zaplněno 28 schránek. Některé

buňky mohou mít více jigů, tím pádem více rolovacích dveří a tlačítek.

dbMailbox_102				
	Name	Data type	Offset	Start value
1	Static			
2	AllData	"udtDbMailbox_20"	0.0	
3	component	Array[1..*iMaxDbM...	0.0	
4	component[1]	"udtMailbox_17"	0.0	
5	iOwner	Int	0.0	0
6	iOrder	Int	2.0	0
7	iStatus	Int	4.0	0

Obrázek 22 – Datový blok Mailbox

V takovém případě se označují jako pozice 1-4. Také může být více robotů. Na následující tabulce je uveden seznam objektů.

Tabulka 5 – Seznam schránek

Schránka	Název	Popis
1	Jig at pos 1	Jig na pozici 1
2	Jig at pos 2	Jig na pozici 2
3	Jig at pos 3	Jig na pozici 3
4	Jig at pos 4	Jig na pozici 4
5	Robot controller 1	Rozvaděč Robot 1
6	Robot controller 2	Rozvaděč Robot 2
7	Robot controller 3	Rozvaděč Robot 3
8	Robot unit 1	Robot 1
9	Robot unit 2	Robot 2
10	Robot unit 3	Robot 3
11	Robot ext axis 1	Externí osa Robot 1
12	Robot ext axis 2	Externí osa Robot 2
13	Robot ext axis 3	Externí osa Robot 3
14	Robot ext axis 4	Externí osa Robot 4
15	Roll door pos 1	Rolovací dveře na pozici 1
16	Roll door pos 2	Rolovací dveře na pozici 2
17	Roll door pos 4	Rolovací dveře na pozici 4
18	Index table	Otočný stůl
19	Index table Aux 1	Pomocná sekvence pro otočný stůl 1
20	Index table Aux 2	Pomocná sekvence pro otočný stůl 2
21	Index table Aux 3	Pomocná sekvence pro otočný stůl 3
22	Index table Aux 4	Pomocná sekvence pro otočný stůl 4
23	Start buttons pos 1	Tlačítko Start na pozici 1
24	Start buttons pos 2	Tlačítko Start na pozici 2
25	Start buttons pos 4	Tlačítko Start na pozici 3
26	Pop-up dialog	Vyskakovací okno na operátorském panelu
27	Quality Check Key sw	Potvrzovací klíč pro kontrolu kvality
28	Code scanner	Snímač čárových kódů

Schránky tedy mají objekty, které vykonávají činnost. K těmto schránkám přistupují řídicí bloky, které jsou označeny jako momentální vlastníci – *iOwner*. Ke každé schránce může najednou přistoupit pouze jeden vlastník, který udává danému objektu příkazy - *iOrder*. Stav o těchto příkazech je uveden v proměnné *iStatus*. Případné doplňovací data - *wExtraDataXX*, jsou používány například pro roboty, jaký jig má svařovat, které hnízdo.

3.1.4 dbNameStrings_26

V tomto datovém bloku se nachází jména k schránkám a k vlastníkům. V datovém bloku *Mailbox* jsou totiž uloženy pouze *Integer* hodnoty, které mají zde přiřazenou hodnotu *String*, tedy slovní popis. Tyto jména jsou čteny, aby při konečném zobrazení byly slovní popisy a ne čísla.

3.2 Vytvořený program

Pro snadnější integraci do existujícího programu byly všechny vytvořené bloky vloženy do složky *SafetyLogging*. Každý funkční blok má svůj přidružený datový blok, kde jsou uloženy používané proměnné v daném funkčním bloku. Za účelem vizualizace byla vytvořena webová stránka a CSV soubor. Tyto možnosti jsou popsány v dalších kapitolách.

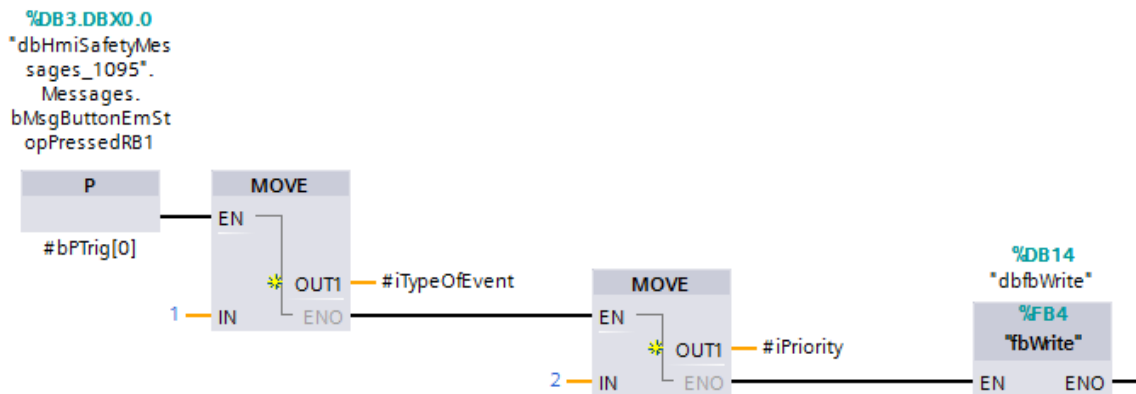


Obrázek 23 – Vytvořené bloky

3.2.1 fbEventsLogging a dbfbEventsLogging

Tento funkční blok je výchozí blok, který se musí spouštět v hlavním organizačním bloku *Main* původního programu, aby se spouštěl v cyklu PLC. Jedná se o první zásah do

původního programu. Dále je zde spuštěn systémový funkční blok *WWW* pro fungování webserveru, vytvořený blok *fbWebServer* a *fbWorkWithCsv*.



Obrázek 24 – Ukázka kódu z *fbEventsLogging*

Hlavní náplní bloku je monitorovat bezpečnostní události. Ty jsou v každém cyklu kontrolovány na náběžnou hranu. Zároveň jsou pomocí proměnné *iTypeOfEvent* identifikovány. Tedy každá událost má svoje unikátní číslo pro pozdější přiřazení správného jména. Zároveň je události přiřazena priorita. Nakonec se volá funkce *fbWrite* pro zaznamenání aktuálních proměnných. Kód obsahuje také pole *bTrig* typu *Bool* o velikosti 112. Toto pole slouží pro systémový blok pro kontrolu náběžné hrany u monitorování událostí.

Poslední funkce v programu je testování proměnné *bEraseDbStatusBlock* a v případě je-li *true*, tak se volá také funkce *fbWrite*. *bEraseDbStatusBlock* slouží k vynulování proměnných v datovém bloku *dbStatusBlock*.

3.2.2 dbStatusBlock

Jedná se datový blok, kam jsou ukládány informace o nastalých bezpečnostních událostech a aktuální hodnoty z datových bloků *dbContext_71*, *dbMailbox_102* při vzniku událostí.

Tento datový blok obsahuje pole o velikosti 100. Jeden element pole je struktura s názvem *StatusBlock* a je to považováno za jeden záznam. Tato struktura obsahuje jméno uložené události - *sTypOfEvent*, čas kdy událost nastala - *ldtTime* a prioritu - *iPriority*. Dále obsahuje všechny data 30 schránek a doplňovací data pro roboty 1, 2 a 3. Poslední jsou data o stavu buňky, stavu objektů a nakládání. Pro vytvoření této struktury musely být převzaty původní struktury pro datové bloky *Context* a *Mailbox*, být upraveny aby

obsahovaly požadované proměnné. Výsledná struktura je tedy vytvořena s dílčími strukturami a proměnnými.

dbStatusBlock				
	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	AllData	"udtStatusDataBloc..		<input checked="" type="checkbox"/>
3	StatusBlock	Array[1..100] of "u...		<input checked="" type="checkbox"/>
4	StatusBlock[1]	"udtStatusDataBlock		<input checked="" type="checkbox"/>
5	sTypeOfEvent	String[50]	"	<input checked="" type="checkbox"/>
6	ldtTime	LDT	LDT#1970-01-01+	<input checked="" type="checkbox"/>
7	iPriority	Int	0	<input checked="" type="checkbox"/>
8	Context	"udtDbEditContext"		<input checked="" type="checkbox"/>
9	MachineStatus	"udtMachineStatus...		<input checked="" type="checkbox"/>
10	ObjectStatus	"udtObjectStatus_2..		<input checked="" type="checkbox"/>
11	bOperatorStationLoad	Bool	TRUE	<input checked="" type="checkbox"/>
12	bOperatorStationUnload	Bool	TRUE	<input checked="" type="checkbox"/>
13	bLoadingContinuesAtNextStation	Bool	false	<input checked="" type="checkbox"/>
14	Mailbox	"udtEditMailbox"		<input checked="" type="checkbox"/>
15	Mailbox	Array[1.."iMaxDbM...		<input checked="" type="checkbox"/>
16	wExtraDataRobot1_01	Word	16#0	<input checked="" type="checkbox"/>
17	wExtraDataRobot1_02	Word	16#0	<input checked="" type="checkbox"/>
18	wExtraDataRobot2_01	Word	16#0	<input checked="" type="checkbox"/>
19	wExtraDataRobot2_02	Word	16#0	<input checked="" type="checkbox"/>
20	wExtraDataRobot3_01	Word	16#0	<input checked="" type="checkbox"/>
21	wExtraDataRobot3_02	Word	16#0	<input checked="" type="checkbox"/>
22	StatusBlock[2]	"udtStatusDataBlock		<input checked="" type="checkbox"/>

Obrázek 25 – Datový blok Status Block

3.2.3 fbWrite a dbfbWrite

Tento vytvořený funkční blok slouží k zápisu dat o události a dat o stavu buňky do datového bloku *dbStatusBlock*. Program je rozdělen na dvě části.

První větší část kódu je dostupná, když proměnná *iTypeOfEvent* je nenulová, což znamená, že je v ní nějaký identifikátor události.

```
#iEr := RD_SYS_T(#ldtTime);

// Writing to variables describe Event Info
#tmpStatusBlock.sTypeOfEvent := #NamesOfEvents["dbfbEventsLogging".iTypeOfEvent];
#tmpStatusBlock.iPriority := "dbfbEventsLogging".iPriority;
#tmpStatusBlock.ldtTime := #ldtTime;
```

Obrázek 26 – Ukládání informací o události

Všechna data, která se zde čtou, aby se uložili do *dbStatusBlock* se ukládají do dočasné *Temp* proměnné *tmpStatusBlock*, který se na konci vloží na správné místo v datovém

bloku. Dále se přes systémovou funkci *RD_SYS_T* čte aktuální čas a ukládá se do proměnné *ldtTime* a zapisují se informace o události.

Dále se zapisují proměnné z datového bloku *Context*.

```
// Writing context
#tmpStatusBlock.Context.bLoadingContinuesAtNextStation := "dbContext_71".AllData.bLoadingContinuesAtNextStation;
#tmpStatusBlock.Context.bOperatorStationLoad := "dbContext_71".AllData.bOperatorStationLoad;
#tmpStatusBlock.Context.bOperatorStationUnload := "dbContext_71".AllData.bOperatorStationUnload;
#tmpStatusBlock.Context.MachineStatus := "dbContext_71".AllData.MachineStatus;
#tmpStatusBlock.Context.ObjectStatus := "dbContext_71".AllData.ObjectStatus;
```

Obrázek 27 – Zapisování Context

Zapisování dat z datového bloku Mailbox je komplikovanější. Ve *for* cyklu se prochází všech 30 schránek a ve vnořeném *for* cyklu se prochází proměnné *OrderStatusName*.

13	OrderStatusName	Struct
14	OrderNumber	Array[0..107] of Int
15	OrderNames	Array[0..107] of String[25]
16	StatusNumber	Array[0..150] of Int
17	StatusNames	Array[0..150] of String[25]

Obrázek 28 – Struktura OrderStatusName

Tato proměnná je struktura obsahující čísla příkazů *iOrder* a statusů *iStatus*. K nim jsou přiřazeny pole s jejich jmény.

Testuje se tedy shoda čísla v *OrderNumber* v struktuře *OrderStatusName* a *iOrder* obsažený ve schránce daného objektu. Až se vyhledá shodné číslo, přiřadí se jméno daného příkazu. Na stejném principu funguje i přiřazení jména statusu.

```
FOR #j := 0 TO #iLengthOrderNames DO
  IF #OrderStatusName.OrderNumber[#j] = "dbMailbox_102".AllData.component[#i].iOrder THEN
    #tmpStatusBlock.Mailbox.Mailbox[#i].sOrder := #OrderStatusName.OrderNames[#j];
    EXIT;
  END_IF;
END_FOR;

FOR #j := 0 TO #iLengthStatusNames DO
  IF #OrderStatusName.StatusNumber[#j] = "dbMailbox_102".AllData.component[#i].iStatus THEN
    #tmpStatusBlock.Mailbox.Mailbox[#i].sStatus := #OrderStatusName.StatusNames[#j];
    EXIT;
  END_IF;
END_FOR;
```

Obrázek 29 – Zapisování jména příkazů a statusů

Jména pro schránku a vlastníka se čtou z datového bloku *dbNameStrings_26*, ta obsahuje struktury *MailboxNames* a *OwnerNames*. Zde odpovídá pořadí schránky pořadí ve strukturách. Proto se testuje jen, jestli schránku někdo vlastní, jinak se zapisuje přímo.

```

IF "dbMailbox_102".AllData.component[#i].iOwner = 0 THEN      // Name of sOwner
    #tmpStatusBlock.Mailbox.Mailbox[#i].sOwner := 'onNone';
ELSE
    #tmpStatusBlock.Mailbox.Mailbox[#i].sOwner := "dbNameStrings_26".AllData.OwnerNames.element["dbMailbox_102".AllData.component[#i].iOwner].sOwnerName;
END_IF;

#tmpStatusBlock.Mailbox.Mailbox[#i].sMailboxName := "dbNameStrings_26".AllData.MailboxNames.component[#i].sMailboxName;    // Name of Mailbox

```

Obrázek 30 – Zapisování jmen názvů schránek a vlastníků

Jako poslední se zapisují doplňovací data robotů 1,2 a 3.

```

// Writing extra data for Robots
#tmpStatusBlock.Mailbox.wExtraDataRobot1_01 := "dbMailbox_102".AllData.component[5].wExtraData01;
#tmpStatusBlock.Mailbox.wExtraDataRobot1_02 := "dbMailbox_102".AllData.component[5].wExtraData02;
#tmpStatusBlock.Mailbox.wExtraDataRobot2_01 := "dbMailbox_102".AllData.component[6].wExtraData01;
#tmpStatusBlock.Mailbox.wExtraDataRobot2_02 := "dbMailbox_102".AllData.component[6].wExtraData02;
#tmpStatusBlock.Mailbox.wExtraDataRobot3_01 := "dbMailbox_102".AllData.component[7].wExtraData01;
#tmpStatusBlock.Mailbox.wExtraDataRobot3_02 := "dbMailbox_102".AllData.component[7].wExtraData02;

```

Obrázek 31 – Zapisování doplňujících dat

Po naplnění celého aktuálního záznamu *tmpStatusBlock* se provádí uložení do *dbStatusBlock*. Po celou dobu je zaznamenáván údaj o současné pozici, kde je uložen poslední záznam - *iCurrentPosition*. Pokud tato pozice překročí maximální hodnotu 100 - *iMaxNuberOfLogs*, začnou se záznamy přepisovat od začátku. Za této situace se ukládá informace, že je blok plně naplněn *bLogsOver100*. Aktuální záznam se tedy uloží na příslušnou pozici a uloží se informace o počtu záznamů. Na konec se provede nulování proměnné *iTypeOfEvent* aby se tento blok programu nevykonával, pokud nedojde k aktivní události.

```

IF #iCurrentPosition + 1 > #iMaxNumberOfLogs THEN
    #iCurrentPosition := 0;
    #bLogsOver100 := true;
END_IF;

#iCurrentPosition := #iCurrentPosition + 1;
"dbStatusBlock".AllData.StatusBlock[#iCurrentPosition] := #tmpStatusBlock;

IF #bLogsOver100 = true THEN
    #iNumberOfLogs := #iMaxNumberOfLogs;
ELSE
    #iNumberOfLogs := #iCurrentPosition;
END_IF;

"dbfbEventsLogging".iTypeOfEvent := 0;

```

Obrázek 32 – Uložení záznamu

Druhá část kódu slouží pro mazání všech dat v *dbStatusBlock*. Podmínka pro vykonání kódu je, aby proměnná *bEraseStatusBlock* byla *true*. Poté se v cyklu nahradí každý záznam

prázdným záznamem a vynulují se hodnoty pro určení pozice posledního záznamu, počet záznamů, samotná proměnná *bEraseStatusBlock* a nastaví se *iInitStatusWS* do režimu 1.

```

IF "dbfbEventsLogging".bEraseDbStatusBlock = true THEN
  FOR #i := 1 TO #iMaxNumberOfLogs DO
    "dbStatusBlock".AllData.StatusBlock[#i] := #EmptyStatusBlock;
  END_FOR;

  #iCurrentPosition := 0;
  #bLogsOver100 := false;
  #iNumberOfLogs := 0;
  "dbfbEventsLogging".bEraseDbStatusBlock := false;
  "dbfbWebServer".iInitStatusWS := 1;
END_IF;

```

Obrázek 33 – Mazání záznamů

3.2.4 fbWebServer a dbfbWebServer

Tento funkční blok slouží k obsluze webové stránky, reaguje na akce uživatele. Webová stránka je popsána později společně se zdrojovým kódem stránky. Zde stačí vědět, že webová stránka má 2 tlačítka pro procházení záznamů – *PREVIOUS* a *NEXT*. A má výběrové menu pro zobrazení konkrétního záznamu. Jedno tlačítko *UPDATE* slouží k aktualizaci statistických proměnných. Stisk tlačítek se projeví jako změna *Bool* proměnných *bNumberUp*, *bNumberDown* a *bStatsStart*. Hodnota z výběrového menu je uložena jako *iNumberWS*. Proměnná *Data* je typu *StatusBlock*, tedy se do ní ukládá jeden záznam z *dbStatusBlock*, podle toho, jaký si uživatel vybral. Pomocí proměnné *iNumber* se ukazuje do pole pro výběr konkrétního záznamu. Proměnná *iNumberShow* slouží k zobrazení čísla procházeného záznamu na webové stránce, důvod zavedení proměnné je popsán v popisu zdrojového kódu stránky.

Celý funkční blok je rozdělen na 5 režimů z důvodu zobrazení dat při prvním načtení stránky. Režimy signalizuje proměnná *iInitStatusWS*.

Režim 0

Hodnota 0 se může vyskytnout pouze po startu programu. Protože proměnná *Data* a i ostatní proměnné jsou po startu nulové, musí dojít k inicializaci proměnných, aby se správně zobrazovaly na webové stránce. Kontroluje se, jestli existuje nějaký záznam pro zobrazení. Pokud neexistuje, režim se přepne do stavu 1, pokud ano, režim je ve stavu 2.

```
IF "dbfbWrite".iNumberOfLogs = 0 THEN
    #iInitStatusWS := 1;
ELSE
    #iInitStatusWS := 2;
END_IF;
```

Obrázek 34 – Obsluha webové schránky, režim 0

Režim 1

Tahle hodnota může být nastavena pouze v režimu 0 nebo při mazání pole záznamů ve funkci *fbWrite*. Aby se nezobrazovali staré hodnoty z proměnné *Data*, je tento blok také vymazán a do proměnné *sTypeOfEvent* je napsána hláška, že datový blok pro události je prázdný. Zároveň je režim přepnut do stavu 3.

```
#Data := "dbfbWrite".EmptyStatusBlock;
#Data.sTypeOfEvent := 'Events data block is empty!';
#iNumberShow := 0;
#iInitStatusWS := 3;
```

Obrázek 35 – Obsluha webové stránky, režim 0

Režim 2

Hodnota tohoto režimu může být nastavena pouze z režimu 0 a 3. Jedná se o situaci, kdy existuje nějaký záznam, a inicializují se data pro první zobrazení. Vždy se zobrazuje první záznam. Do proměnné *Data* se ukládá první záznam z pole záznamů, hodnoty *iNumber* a *iNumberShow* jsou nastaveny do 1. Režim je přepnuto do stavu 4.

```
#Data := "dbStatusBlock".AllData.StatusBlock[1];
#iNumberShow := 1;
#iNumber := 1;
#iInitStatusWS := 4;
```

Obrázek 36 – Obsluha webové stránky, režim 2

Režim 3

Do tohoto režimu se lze dostat z režimu 1.

```
IF "dbfbWrite".iNumberOfLogs > 0 THEN
    #iInitStatusWS := 2;
END_IF;
#bNumberDown := false;
#bNumberUp := false;
```

Obrázek 37 – Obsluha webové stránky, režim 3

Zde se pouze čeká, až bude existovat aspoň jeden záznam, to poté dochází ke změně

režimu na hodnotu 2. Pokud stále není žádný záznam, nulováním proměnných přiřazeným k tlačítkům se ignoruje jejich stlačení.

Režim 4

Režim se spouští jen z režimu 2. Jedná se o hlavní část, kde se monitorují zvlášť tlačítka a zvlášť výběrové menu. Pokud dojde ke stisku některého tlačítka, tak pokud je to *NEXT*, tedy proměnná *bNumberUp* je *true*, zvýší se proměnná *iNumber* o 1. Kontroluje se zda nová zvýšená hodnota nepřekročila počet záznamů, pokud ano, zobrazí se první záznam, tedy procházení pokračuje z posledního záznamu na první. Proměnná *bNumberUp* se nuluje.

Podobně je to i s proměnnou *iNumberDown* při její aktivaci se ale *iNumber* snižuje o -1 a kontroluje se, zda není menší než 1, v tom případě pokračuje z prvního záznamu na poslední. Proměnná *bNumberDown* se nuluje. Nakonec dochází ke kopírování vybraného záznamu do *Data*. Do proměnné *iNumberShow* se zapisuje aktuální hodnota *iNumber*.

```
IF #bNumberUp = true OR #bNumberDown = true THEN

    IF #bNumberUp = true THEN
        #iNumber := #iNumber + 1;
        IF #iNumber > "dbfbWrite".iNumberOfLogs THEN
            #iNumber := 1;
        END_IF;
        #bNumberUp := false;
    END_IF;

    IF #bNumberDown = true THEN
        #iNumber := #iNumber - 1;
        IF #iNumber < 1 THEN
            #iNumber := "dbfbWrite".iNumberOfLogs;
        END_IF;
        #bNumberDown := false;
    END_IF;

    #Data := "dbStatusBlock".AllData.StatusBlock[#iNumber];
    #iNumberShow := #iNumber;
END_IF;
```

Obrázek 38 – Obsluha webové stránky, režim 4, tlačítka

Ve stejném režimu se kontroluje, zda číslo *iNumberWS* je v rozsahu 1 až počet záznamů. Pokud ano, přepíše se hodnota do *iNumber* a *iNumberShow*, do *Data* se zkopíruje vybraný záznam a *iNumberWS* se vynuluje, aby nedocházelo v příštím cyklu k opětovnému

vykonání. Rozsah *iNumberWS* se musí kontrolovat z důvodu, že hodnota se předává přes URL adresu a uživatel by mohl dopsat jakékoliv číslo a způsobit tím chybu.

```

IF #iNumberWS > 0 AND #iNumberWS <= "dbfbWrite".iNumberOfLogs THEN
    #iNumber := #iNumberWS;
    #Data := "dbStatusBlock".AllData.StatusBlock[#iNumber];
    #iNumberWS := 0;
    #iNumberShow := #iNumber;
END_IF;

```

Obrázek 39 – Obsluha webové stránky, režim 4, výběrové menu

Jako poslední se reaguje na proměnnou *bStatsStart*. Při její aktivaci se volá funkce *fbEventsStats* a proměnná se nuluje.

```

IF #bStatsStart = true THEN
    "dbfbEventsStats"();
    #bStatsStart := false;
END_IF;

```

Obrázek 40 – Spuštění funkce pro statistiku

3.2.5 fbEventsStats a dbfbEventsStats

Tento blok slouží k základním statistickým vyhodnocením vzniklých událostí. Hodnoty uvádějící stáří událostí jsou uloženy v *statsArrayTime* a hodnoty pro statistiku priorit v *statsArrayPriority*. Kontroluje se, zda existují nějaké záznamy, pokud ne, všechny proměnné se nastaví do nuly.

```

IF "dbfbWrite".iNumberOfLogs = 0 THEN
    FOR #j := 0 TO #iNumberOfLimits DO
        #statsArrayTime[#j] := 0;
        #statsArrayPriority[#j] := 0;
    END_FOR;
ELSE

```

Obrázek 41 – Nulování statistických hodnot

Pokud existují nějaké záznamy, čte se aktuální čas *ldtCurrentTime* a v cyklu *for* se vytváří časový rozdíl vzniklého záznamu a aktuálního času - *ltimeDifference* pomocí systémové funkce *T_DIFF*. Následně se ve vnořeném cyklu *for* kontroluje rozmezí, v jakém záznam vznikl. Rozmezí jsou 0 až 1 den, 1den až 1 týden, 1 týden až 1 měsíc, 1 měsíc až 6 měsíců, 6 měsíců až 1 rok a starší. Tyto limity jsou uloženy v poli *timeLimits*. Pro zjištění četnosti priorit 1, 2 a 3 se cyklu procházení záznamů kontroluje daná priorita a podle *CASE* funkce a se inkrementuje daná proměnná značící četnost priority.

```

#iError := RD_SYS_T(#ldtCurrentTime);
FOR #i := 1 TO "dbfbWrite".iNumberOfLogs DO
  // time stats
  #ltimeDifference := T_DIFF(IN1 := #ldtCurrentTime, IN2 := "dbStatusBlock".AllData.StatusBlock[#i].ldtTime);

  FOR #j := 0 TO #iNumberOfLimits DO

    IF #ltimeDifference > #timeLimits[#j] AND #ltimeDifference < #timeLimits[#j+1] THEN
      #statsArrayTime[#j] := #statsArrayTime[#j] + 1;
    END_IF;

  END_FOR;

  // priority stats
  CASE "dbStatusBlock".AllData.StatusBlock[#i].iPriority OF
    1:
      #statsArrayPriority[0] := #statsArrayPriority[0] + 1;
    2:
      #statsArrayPriority[1] := #statsArrayPriority[1] + 1;
    3:
      #statsArrayPriority[2] := #statsArrayPriority[2] + 1;
  END_CASE;
END_FOR;

```

Obrázek 42 – Statistické vyhodnocení záznamů

3.2.6 fbWorkWithCsv a dbfbWorkWithCsv

Tento blok také monitoruje tlačítka webové stránky. Jsou to dvě tlačítka. Tlačítko *CREATE*, slouží k vytvoření CSV souboru a je provázáno s proměnnou *bCreateStart*. Tlačítko *DELETE*, slouží k smazání CSV souboru a je provázáno s proměnnou *bDeleteStart*. Systémové funkce pro práci s CSV soubory jsou složité a práce s nimi komplikovaná, proto jsou ve zvláštním funkčním bloku.

Pro CSV soubor se dá vytvořit hlavička, která obsahuje názvy pro sloupce. Ty se budou skládat z názvu všech proměnných v jednom záznamu, což je 170 proměnných. Jeden *String* má délku 256, což je málo pro pokrytí všech názvů, proto bylo vytvořeno pole *nameStrings* s 11 prvky typu *String*.

```

IF #bConvert = false THEN

  FOR #i := 0 TO 10 DO
    Strg_TO_Chars(Strg := #nameStrings[#i],
                  pChars := #iPosition,
                  Cnt => #iNumberOfChars,
                  Chars := #header);
    #iPosition := #iPosition + #iNumberOfChars;
  END_FOR;

  #bConvert := true;
END_IF;

```

Obrázek 43 – Tvorba hlavičky

Do těchto prvků byly napsány všechny názvy proměnných, navíc každý název musí být oddělen čárkou, což slouží v CSV souboru jako oddělovač sloupců. Toto pole bylo poté převedeno na pole typu *Char* o délce 2002 znaků s názvem *header*. Kód se provádí jen jednou po startu PLC pomocí proměnné *bConvert*.

Pro vytvoření CSV souboru slouží systémová funkce *DataLogCreate*. Tato funkce obsahuje několik vstupních i výstupních proměnných:

REQ – pro spuštění funkce

RECORDS – maximální počet záznamů

TIMESTAMP – pro zobrazování času přidání řádku

NAME – jméno CSV souboru

DONE – vytvoření proběhlo úspěšně

BUSY – probíhá vytváření

ERROR – akce skončila chybou

STATUS – hodnoty ukazují stav po ukončení funkce, případně druh chyby

ID – jedinečné identifikační číslo

HEADER – pro vytvoření hlavičky

DATA – data pro vložení

Funkce je spouštěna pomocí náběžné hrany *bCreateStart* z webové stránky. Všem potřebným parametrům funkce jsou přiřazeny proměnné. Maximální počet záznamů je 100, jméno souboru je vždy stejné *EventsDataLog.csv*.

```
"dbR_TRIGS_DataLogCreateSafetyEvents" (CLK:=#bCreateStart,  
Q=>#bCreateStartP);  
  
"dbDataLogCreateSafetyEvents" (REQ:=#bCreateStartP,  
RECORDS:=100,  
TIMESTAMP:=0,  
NAME:=#sName,  
DONE=>#bCdone,  
BUSY=>#bCbusy,  
ERROR=>#bCerror,  
STATUS=>#wCstatus,  
ID:=#dwMyid,  
HEADER:=#header,  
DATA:=#Data);
```

Obrázek 44 – Vytvoření csv souboru, hlavní funkce

Všechny funkce obsluhující CSV soubor musí stále cyklovat. Proměnné přiřazené parametrům mění své hodnoty každým cyklem, znamená to, že pokud byl soubor CSV vytvořen, tak *bCdone* bude *true* jen jeden cykl. Stejně tak *bCerror* a *wCstatus*. Proto je

potřeba tyto hodnoty ukládat do pomocných proměnných, aby se uchovaly jejich hodnoty i pro další cykly. Pokud došlo k vytvoření souboru, nastavuje se hodnota *true* do *bCdone1* a *bWdone1*, aby mohl být zahájen zápis dat. Pokud došlo k chybě, zapisuje se do pomocné proměnné *wCstatus1* druh chyby a nulováním startovací proměnné se umožní další pokus o vytvoření. Pokud druh chyby je *16#8093*, značí to, že byl příkaz pro vytvoření souboru, ale ten už existuje. Nastavuje se proto *true* do *bDeleteStart* pro smazání souboru a *bDelayedStart* pro automatické vytvoření souboru po jeho smazání.

```
IF #bCdone = true THEN
    #bCdone1 := true;
    #bWdone1 := true;
END_IF;

IF #bCerror = true THEN
    #wCstatus1 := #wCstatus;
    #bCreateStart := false;
END_IF;

IF #wCstatus1 = 16#8093 THEN
    #bDeleteStart := true;
    #bDelayedStart := true;
END_IF;
```

Obrázek 45 – Vytvoření csv souboru, pomocné proměnné

Pro zapsání do souboru slouží systémová funkce *DataLogWrite*. Její parametry jsou podobné jako u funkce pro vytvoření. Na jedno zavolání funkce lze zapsat jeden řádek do souboru. Proto při potřebě zapsat více řádku se musí postupně, jeden po druhém. Jak vytvoření souboru, tak mazání i zápis může trvat několik cyklů PLC. Proto nelze použít pro zapisování *for*.

Testují se proměnné *bCdone1*, což zaručuje, že je vytvořený soubor a *bWdone1*, který určuje, že byl úspěšně zapsán předešlý záznam. Pak je aktivní proměnná *bWriteStart* pro spuštění funkce pro zapisování. *iCounter* slouží jako počítadlo, které počítá zapsané záznamy a ukazuje od pole záznamů v *dbStatusBlock*. Pokud je hodnota menší, než počet záznamů, zapisují se data z pole záznamů do proměnné *Data* (na následujícím obrázku pod příkazem *ELSE*, ale kvůli úspoře místa nebyl kód zobrazen) a dochází k zápisu do souboru. Pokud počítadlo překročí velikost počtu záznamů, došlo k zápisu všech záznamů a zápis musí být zastaven. Nuluje se tedy startovací proměnná pro zápis i proměnná potvrzující vytvoření souboru a počítadlo. Nulováním proměnné *bCreateStart* se umožní opětovné vytvoření souboru.

```

IF #bWdone1 = 1 AND #bCdone1 = true THEN
    #bWriteStart := true;
    #iCounter := #iCounter + 1;

    IF #iCounter > "dbfbWrite".iNumberOfLogs THEN
        #bCdone1 := false;
        #bWriteStart := false;
        #bCreateStart := false;
        #iCounter := 0;
    ELSE

```

Obrázek 46 – Podmínky a logika pro zápis dat do souboru

Data ukládající se do souboru může být struktura, ale uvnitř už nesmí být další struktury nebo pole, tedy nelze vložit strukturu *StatusBlock*. Proto pro proměnnou *Data* byla vytvořena struktura *udtDataLog*, která obsahuje 170 proměnných, každá samostatně na jednom řádku. Proto i zápis musel být proveden pro každou proměnnou zvlášť, jak je zobrazeno na následujícím obrázku.

```

#Data.M1_sStatus := "dbStatusBlock".AllData.StatusBlock[#iCounter].Mailbox.Mailbox[1].sStatus;
#Data.M2_sName := "dbStatusBlock".AllData.StatusBlock[#iCounter].Mailbox.Mailbox[2].sMailboxName;
#Data.M2_sOwner := "dbStatusBlock".AllData.StatusBlock[#iCounter].Mailbox.Mailbox[2].sOwner;
#Data.M2_sOrder := "dbStatusBlock".AllData.StatusBlock[#iCounter].Mailbox.Mailbox[2].sOrder;

```

Obrázek 47 – Ukázka zápisu dat do Data

Funkce pro zapsání dat od souboru má také podobné parametry jako funkce pro vytvoření.

```

"dbDataLogWriteSafetyEvents" (REQ:=#bWriteStart,
    DONE=>#bWdone,
    BUSY=>#bWbusy,
    ERROR=>#bWerror,
    STATUS=>#wWstatus,
    ID:=#dwMyid);

```

Obrázek 48 – Funkce pro zapsání dat do souboru

Stejně jako funkce pro vytvoření souboru i tady se musí výstupy funkce zapisovat do pomocných proměnných. Pomocná proměnná *bWdone1* slouží jako jeden ze spouštěčů pro zápis dat. Pokud dojde k chybě, uloží se druh chyby a nulují se startovací proměnné pro umožnění dalšího pokusu o vytvoření souboru.

```

IF #bWdone = true THEN
    #bWdone1 := true;
ELSE
    #bWdone1 := false;
END_IF;

IF #bWerror = true THEN
    #wWstatus1 := #wWstatus;
    #bCdone1 := false;
    #bWriteStart := false;
    #bCreateStart := false;
    #iCounter := 0;
END_IF;

```

Obrázek 49 – Zapisování dat, pomocné proměnné

Poslední možnost pro práci se souborem je smazání souboru. K tomu slouží systémová proměnná *DataLogDelete*. Nový parametry je *DelFile* – pro úplné odstranění souboru. Může se mazat podle *ID* nebo podle jména. Pokud je *ID* nulové, maže se podle jména, jinak má *ID* přednost.

Pokud je náběžná hrana *bDeleteStart true* dochází k aktivaci mazání souboru.

```

"dbR_TRIGS_DataLogDeleteSafetyEvents" (CLK=#bDeleteStart,
    Q=>#bDeleteStartP);

"dbDataLogDeleteSafetyEvents" (REQ=#bDeleteStartP,
    NAME=#sName,
    DelFile:=true,
    DONE=>#bDdone,
    BUSY=>#bDbusy,
    ERROR=>#bDerror,
    STATUS=>#wDstatus,
    ID:=#dwDid);

```

Obrázek 50 – Funkce pro smazání csv souboru

Stejně jako ostatní funkce i zde musí být zavedeny pomocné proměnné. Pokud dojde k nějaké chybě, je uložen druh chyby a nulováním startovací proměnné je umožněn další pokus, to se může stát, pokud je stisknuto tlačítko pro mazání, ale žádný soubor neexistuje. Pokud dojde k úspěšnému smazání, je umožněno další spuštění, ale také se kontroluje, zda není sepnutý opožděný start vytvoření souboru, pokud ano, startovací proměnná pro vytvoření souboru je aktivována, opožděný start vypnut a nulován status o vytvoření.

```

IF #bDerror = true THEN
    #wDstatus1 := #wDstatus;
    #bDeleteStart := false;
END_IF;

IF #bDdone = true THEN
    #bDeleteStart := false;

    IF #bDelayedStart = true THEN
        #bCreateStart := true;
        #bDelayedStart := false;
        #wCstatus1 := 0;
    END_IF;
END_IF;

```

Obrázek 51 – Pomocné proměnné funkce pro mazání csv souboru

3.2.7 Zdrojové soubory webové stránky

Vytvořené soubory pro chod webové stránky se nahrávají do PLC a ukládají se do datových bloků, kde zabírají místo v paměti, proto byla snaha vytvořit tyto soubory co nejmenší bez použití dodatečných knihoven, pouze za pomoci kódu HTML, kaskádových stylů a *java scriptu*.

Jelikož firma už využívala možnost uživatelsky vytvořené stránky, muselo dojít k druhému zásahu do původního kódu a musela být jejich stránka upravena přidáním tlačítka pro přesměrování na stránku pro procházení záznamů. Do původní složky *Web2PLCInitial* byla vložena složka *SafetyLogs*. Ta obsahuje složku *CSS* pro zdrojový soubor style s kaskádovými styly, složku *js* se souborem *core* pro *java script*. A HTML soubor *safety* s HTML kódem.

```

<!-- AWP_In_Variable Name=' "dbfbWebServer".bNumberDown' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".bNumberUp' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".bStatsStart' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".iNumberWS' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".data' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".iNumberShow' -->
<!-- AWP_In_Variable Name=' "dbfbWebServer".iNumberOfLogsShow' -->
<!-- AWP_In_Variable Name=' "dbfbWorkWithCsv".bCreateStart' -->
<!-- AWP_In_Variable Name=' "dbfbWorkWithCsv".bDeleteStart' -->
<!-- AWP_In_Variable Name=' "dbfbEventsStats".statsArrayPriority' -->
<!-- AWP_In_Variable Name=' "dbfbEventsStats".statsArrayTime' -->

```

Obrázek 52 – AWP příkazy

Pro zobrazení reálných dat uložených v PLC je zapotřebí speciálních příkazů, odkazujících na data, které chceme číst. Jedná se o AWP (*Automation Web Programming*) příkazy.

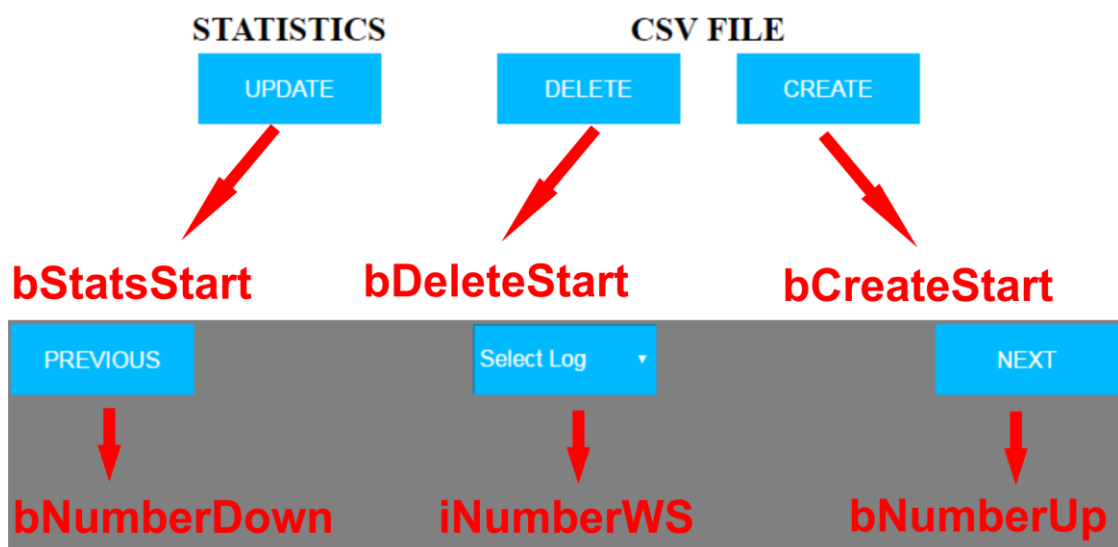
Tyto příkazy obsahují cestu k datům, která chceme číst, či do nich zapisovat. Pokud označíme celý datový blok, lze přistupovat ke všem datům, a jelikož se data přenáší odkazem ve webovém prohlížeči, je to z hlediska bezpečnosti nepřijatelné. Proto jsou použity odkazy na konkrétní proměnné nebo na datové bloky, kde data jen čteme a jejich úpravou by se nic nestalo. Například proměnná *iNumberShow* odpovídá proměnné *iNumber*, která ukazuje do pole záznamů. Její hodnota se posílá přes URL adresu, proto by mohla být změněna na libovolné číslo a mohla ukazovat do neexistujícího záznamu a tím by došlo k zastavení PLC. Proto se číslo posílá přes pomocnou proměnnou, u které se poté hodnota kontroluje a až potom přiřazuje proměnné *iNumber*.

Pro přístup k proměnné, kterou chceme číst, slouží HTML tagy. V programu jsou tyto tagy použity v *java scriptu*, kde se celé přiřazují proměnné. Bohužel není možné tyto tagy generovat v cyklu, protože takovýto kód by při kompilaci v PLC při převodu do datových bloků nebyl rozeznán, jelikož jejich podoba je vytvořena až v prohlížeči. Proto každá proměnná musí být přiřazena samostatně na řádku, jak je na následujícím obrázku.

```
TypeOfEvent = ':= "dbfbWebServer".Data.sTypeOfEvent: ';
Time = ':= "dbfbWebServer".Data.ldtTime: ';
Priority = ':= "dbfbWebServer".Data.iPriority: ';
NumberOfEvent = ':= "dbfbWebServer".iNumberShow: ';
```

Obrázek 53 – Čtení dat z PLC

Proto je vždy možné procházet pouze jen jeden záznam, podle toho, jaký si uživatel zvolí. V *java scriptu* se tedy načítají proměnné, upravují se datové typy *String*, protože se z PLC načítají i s uvozovkami a vytváří se tabulky.



Obrázek 54 – Provázaní tlačítek a proměnných

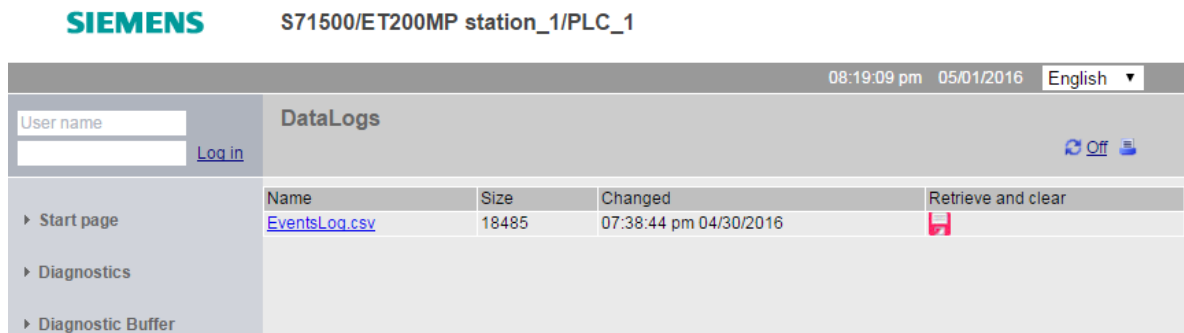
Webový server je vytvořen tak, že každé tlačítko je provázáno s jistou proměnnou a jeho stiskem dojde k nastavení proměnné na *true*. Tato změna se vyhodnotí už v PLC a tam se provede potřebná akce. Výběrové menu je typu *Integer*.

3.3 Výstupní zobrazení

Pro výstup uložených dat byly zvoleny dva způsoby – CSV soubor a webová stránka. Přístup k těmto věcem je pomocí webserveru popsán v kapitole 1.3.4

3.3.1 CSV soubor

Pro přístup k CSV souboru je potřeba na webserveru vybrat položku *DataLogs*, pak se kliknutím na existující soubor stáhne do počítače a lze ho otevřít v programu *Excel*.



Obrázek 55 – CSV soubor na webserveru

Tento soubor slouží především k celkovému pohledu na všechny zaznamenané události zároveň. Také je určen k seřazení podle aktuálně potřebného kritéria. Pro hlubší analýzu jednoho záznamu může být při větším počtu záznamů matoucí, k tomu je určena webová stránka.

	A	B	C	D	E	F	G	H	I
1	SeqNo	Type of Event	Time	Priority	MS-bStop	MS-breset	MS-bStart	MS-Servic	MS-bAuto
2	1	bMsgOperatorPos1HorizontalFloorAreaErrorDiscFault	2016-05-02 10:32:16.605_747_199	3	0	0	0	0	1
3	2	bMsgButtonEmStopPressedRB3	2016-05-02 10:36:55.956_479_902	3	0	1	0	0	0
4	3	bMsgFeedbackMC_Indextable_CCW	2016-05-02 10:36:55.956_738_995	3	0	1	0	0	0
5	4	bMsgFeedbackMC_Indextable_CCW_CHA	2016-05-02 10:36:58.306_203_279	3	0	1	0	0	0
6	5	bMsgFeedbackTooling4	2016-05-02 10:41:09.239_103_995	3	0	0	0	0	1
7	//END								

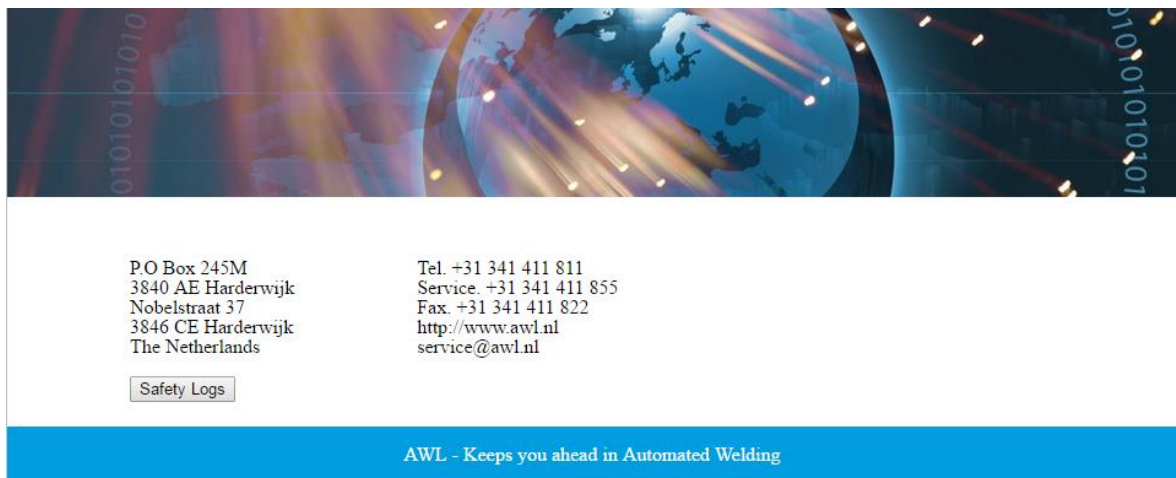
Obrázek 56 – Ukázka csv souboru

Soubor obsahuje v prvním řádku hlavičku s názvy proměnných. Každý sloupec je jedna proměnná. Každý řádek znamená jeden záznam aktivních bezpečnostních událostí z pole *StausBlock*.

3.3.2 Webová stránka

Primárně slouží webová stránka pro hlubší analýzu jednoho záznamu. Jak už bylo popsáno výše, není možné zobrazit všechny záznamy zároveň, tomu slouží CSV soubor. Pro přístup k stránce se musí na webserveru vybrat položka *CostumerPages* a *Homepage of the application*.

Zobrazí se úvodní stránka firmy s kontakty, kde bylo doplněno tlačítko pro přesměrování na vytvořenou stránku pro procházení záznamů bezpečnostních událostí.



Obrázek 57 – Původní webová stránka

Vytvořená webová stránka se skládá z 5 částí:

1. *Event Info*

Zde se nachází informace o aktuálně prohlíženém záznamu a informaci o celkovém počtu záznamů. Dále jméno události, čas vzniku a priorita.

2. *CSV File*

Jedná se o oblast, kde se dá generovat nebo mazat CSV soubor. Pokud dojde k požadavku na vytvoření souboru a soubor už existuje, starý se automaticky smaže a vytvoří se soubor nový. Pokud dojde k požadavku na smazání a soubor neexistuje, tak se tento příkaz ignoruje.

3. *Statistics*

Znázorněna je zde základní statistika událostí. První je rozdělení podle doby, kdy událost vznikla. Období jsou rozděleny na stáří: 1 den, 1 týden, 1 měsíc, půl roku, rok a starší. Další rozdělení je podle priorit, zobrazuje se tedy četnost priorit

existujících záznamů. Tlačítkem dochází k aktualizaci těchto proměnných, jinak zůstávají zobrazeny hodnoty při poslední aktualizaci. Je to z důvodu, že pokud chce někdo vidět aktuální statistické data, potřebuje to vědět vztažené pro danou chvíli a ne například den staré.

Safety Events Logs

EVENT INFO

Event: 1 of 5

Name: bMsgOperatorPos1HorizontalFloorAreaErrorDiscFault

Time: 2016-02-10:32:16.60

Priority: 3

CSV FILE

DELETE
CREATE

Age of events

1 Day	1 Week	1 Month	6 Months	1 Year	Older
5	0	0	0	0	0

Number of priorities

Priority 1	Priority 2	Priority 3
0	0	5

STATISTICS

UPDATE

MACHINE STATUS

Machine Status
bStop
bReset
bStart
bService
bAuto
bDelayedStop
bDelayedReset
bDelayedStart
bDelayedService

MAILBOXES

Mailbox	Owner	Order	Status
1: Jig at pos 1	takeout/inlay pos 1	odJigLoadProducts	stProcessing
2: Jig at pos 2	process station side	odNone	stReady
3: Jig at pos 3	onNone	odNone	stNotInitialized
4: Jig at pos 4	onNone	odNone	stNotInitialized
5: Robot controller 1	process station side	odRobotWeld	stProcessing
6: Robot controller 2	process station side	odRobotWeld	stProcessing
7: Robot controller 3	onNone	odNone	stNotInitialized
8: Robot unit 1	onNone	odNone	stReady
9: Robot unit 2	onNone	odNone	stReady

Obrázek 58 – Webová stránka, horní část

4. Machine Status

Zobrazeny jsou zde data z datového bloku *dbContext_71*, tedy informace popisující stav buňky. Jedná se o proměnné typu *Bool*, pokud jsou proměnné *true*, jsou podbarveny zeleně, pokud jsou *false*, jsou podbarveny červeně. Jedna proměnná je typu *Integer*, proto je vepsáno číslo k jménu proměnné.

5. Mailboxes

Zde se nachází hodnoty z datového bloku *dbMailbox_102*. Zobrazeno je jméno schránky, vlastník, příkaz a status daného příkazu. Spodní tabulka ukazuje doplňovací data pro roboty 1, 2 a 3. Zároveň obsahuje navigační tlačítka pro

procházení záznamů pro další a předešlý záznam. Nebo pomocí výběrového menu se dá přejít na konkrétní záznam.

bPowerOn
bOneOrMoreTeachActive
bLastScanHadMessage
bLastScanHadFailure
Object Status
bResetNecessary
bStatusChanged
bCycleStopNecessary
bMessage
bFailure
bAckFlash
bSignalsAreSimulated
bFatalFailureForceStop
bRequestStop
bTeachActive
iInternalProgramError: 0
bOperatorStationLoad
bOperatorStationUnload
bLoadingContinuesAtNextStation

25: Start buttons pos 4	onNone	odNone	stNotInitialized
26: Pop-up dialog	onNone	odNone	stReady
27: Quality Check Key sw	onNone	odNone	stReady
28: Code scanner	takeout/inlay pos 1	odStart	stProcessing
29:	onNone	odNone	stNotInitialized
30:	onNone	odNone	stNotInitialized
Extra Data	Value 01	Value 02	
Robot 1	3	0	
Robot 2	3	0	
Robot 3	0	0	

Navigation: PREVIOUS | Select Log | NEXT

AWL - Keeps you ahead in Automated Welding

Obrázek 59 – Webová stránka, dolní část

Při procházení záznamů dochází k rotaci při směru dopředu z posledního záznamu na první a při směru dozadu z prvního na poslední. V menu se objeví jen tolik záznamů, kolik existuje. Pokud nejsou žádné záznamy, zobrazí se stránka s hláškou, že datový blok záznamů je prázdný a stisky tlačítek jsou v PLC ignorovány.

Jak už bylo uvedeno, maximální počet záznamů je určen na 100, pak se záznamy podle stáří začnou přepisovat. Tento počet byl stanoven s ohledem na to, že nová buňka by neměla hlásit žádné události. Pokud se událost spolu s potvrzením vyskytne jednou za den, tak by záznam měl zůstat uložený 50 dnů, pokud se denně vyskytne více událostí, tak může být ohrožena funkčnost buňky a servisní technik se do několika dnů dostaví na servis a data zůstanou také uložena. Pokud dojde k zastavení provozu buňky, tak ani nové události nevznikají a tedy jsou data k dispozici. Tyto navržené časové úseky byly firmou schváleny.

3.3.3 Ukázka dat

Pro ukázkou jsou zobrazeny dva záznamy s reálnými daty při dvou stavech. Při automatickém chodu buňky a při restartu buňky.

MACHINE STATUS	MAILBOXES			
Machine Status	Mailbox	Owner	Order	Status
bStop	1: Jig at pos 1	takeout/inlay pos 1	odJigLoadProducts	stProcessing
bReset	2: Jig at pos 2	process station side	odNone	stReady
bStart	3: Jig at pos 3	onNone	odNone	stNotInitialized
bService	4: Jig at pos 4	onNone	odNone	stNotInitialized
bAuto	5: Robot controller 1	process station side	odRobotWeld	stProcessing
bDelayedStop	6: Robot controller 2	process station side	odRobotWeld	stProcessing
bDelayedReset	7: Robot controller 3	onNone	odNone	stNotInitialized
bDelayedStart	8: Robot unit 1	onNone	odNone	stReady
bDelayedService	9: Robot unit 2	onNone	odNone	stReady
bDelayedAuto	10: Robot unit 3	onNone	odNone	stNotInitialized
bEMS_OK	11: Robot ext axis 1	onNone	odNone	stReady
bEMS_OK_Toff	12: Robot ext axis 2	process station side	odRobotWeld	stProcessing
bFence_OK	13: Robot ext axis 3	onNone	odNone	stNotInitialized
bFence_Open	14: Robot ext axis 4	onNone	odNone	stNotInitialized
bEnable	15: Roll door pos 1	takeout/inlay pos 1	odNone	stDoorOpened
bAck	16: Roll door pos 2	onNone	odNone	stNotInitialized
bCycleStopActive	17: Roll door pos 4	onNone	odNone	stNotInitialized
bFirstScan	18: Index table	Rotation logistics	odNone	stReady
bSimulate	19: Index table Aux 1	onNone	odNone	stNotInitialized
bCommissioning	20: Index table Aux 2	onNone	odNone	stNotInitialized
bLightTest	21: Index table Aux 3	onNone	odNone	stNotInitialized
bDryRun	22: Index table Aux 4	onNone	odNone	stNotInitialized
bIsStoppedDirectly	23: Start buttons pos 1	takeout/inlay pos 1	odWaitForPosEdge	stProcessing
bPowerOn	24: Start buttons pos 2	onNone	odNone	stNotInitialized
bOneOrMoreTeachActive	25: Start buttons pos 4	onNone	odNone	stNotInitialized
bLastScanHadMessage	26: Pop-up dialog	onNone	odNone	stReady
bLastScanHadFailure	27: Quality Check Key sw	onNone	odNone	stReady
Object Status	28: Code scammer	takeout/inlay pos 1	odStart	stProcessing
bResetNecessary	29:	onNone	odNone	stNotInitialized
bStatusChanged	30:	onNone	odNone	stNotInitialized
bCycleStopNecessary	Extra Data	Value 01	Value 02	
bMessage	Robot 1	3	0	
bFailure	Robot 2	3	0	
bAckFlash	Robot 3	0	0	
bSignalsAreSimulated				
bFatalFailureForceStop				
bRequestStop				
bTeachActive				
nInternalProgramError: 0				
bOperatorStationLoad				
bOperatorStationUnload				
bLoadingContinuesAtNextStation				

Obrázek 60 – Ukázka dat při automatickém režimu

MACHINE STATUS	MAILBOXES			
Machine Status	Mailbox	Owner	Order	Status
bStop	1: Jig at pos 1	Reset jigs logistics	odLimitSoftware	stProcessing
bReset	2: Jig at pos 2	Reset jigs logistics	odLimitSoftware	stProcessing
bStart	3: Jig at pos 3	onNone	odNone	stNotInitialized
bService	4: Jig at pos 4	onNone	odNone	stNotInitialized
bAuto	5: Robot controller 1	process station side	odLimitSoftware	stProcessing
bDelayedStop	6: Robot controller 2	process station side	odLimitSoftware	stProcessing
bDelayedReset	7: Robot controller 3	onNone	odNone	stNotInitialized
bDelayedStart	8: Robot unit 1	onNone	odNone	stReady
bDelayedService	9: Robot unit 2	onNone	odNone	stReady
bDelayedAuto	10: Robot unit 3	onNone	odNone	stNotInitialized
bEMS_OK	11: Robot ext axis 1	onNone	odNone	stReady
bEMS_OK_Toff	12: Robot ext axis 2	onNone	odNone	stReady
bFence_OK	13: Robot ext axis 3	onNone	odNone	stNotInitialized
bFence_Open	14: Robot ext axis 4	onNone	odNone	stNotInitialized
bEnable	15: Roll door pos 1	takeout/inlay pos 1	odLimitHardware	stProcessing
bAck	16: Roll door pos 2	onNone	odNone	stNotInitialized
bCycleStopActive	17: Roll door pos 4	onNone	odNone	stNotInitialized
bFirstScan	18: Index table	Rotation logistics	odLimitHardware	stProcessing
bSimulate	19: Index table Aux 1	onNone	odNone	stNotInitialized
bCommissioning	20: Index table Aux 2	onNone	odNone	stNotInitialized
bLightTest	21: Index table Aux 3	onNone	odNone	stNotInitialized
bDryRun	22: Index table Aux 4	onNone	odNone	stNotInitialized
bIsStoppedDirectly	23: Start buttons pos 1	takeout/inlay pos 1	odNone	stReady
bPowerOn	24: Start buttons pos 2	onNone	odNone	stNotInitialized
bOneOrMoreTeachActive	25: Start buttons pos 4	onNone	odNone	stNotInitialized
bLastScanHadMessage	26: Pop-up dialog	onNone	odNone	stReady
bLastScanHadFailure	27: Quality Check Key sw	onNone	odNone	stReady
Object Status	28: Code scanner	takeout/inlay pos 1	odNone	stReady
bResetNecessary	29:	onNone	odNone	stNotInitialized
bStatusChanged	30:	onNone	odNone	stNotInitialized
bCycleStopNecessary	Extra Data	Value 01	Value 02	
bMessage	Robot 1	3	0	
bFailure	Robot 2	3	0	
bAckFlash	Robot 3	0	0	
bSignalsAreSimulated	PREVIOUS	Select Log	NEXT	
bFatalFailureForceStop				
bRequestStop				
bTeachActive				
iInternalProgramError: 0				
bOperatorStationLoad				
bOperatorStationUnload				
bLoadingContinuesAtNextStation				

Obrázek 61 – Ukázka dat při restartu

Při automatickém chodu jsou aktivní proměnné *bAuto* a *bDelayedAuto*, což značí aktuální režim. *bEMS_OK* značí, že není aktivní žádné nouzové stop tlačítko. Že jsou zavřeny servisní dveře, ukazuje *bFence_OK*. *bEnable* značí aktivaci logické stanice pro proces. *bPowerOn* ukazuje, že je buňka napájena a připravena k akci. *bOperatorStationLoad* a

bOperatorSrationUnload značí nakládání a vykládání produktů. Schránku *Jig at Pos 1* vlastní blok pro pozice operátora 1 a značí náklad/výklad produktu. *Jig at Pos 2* vlastní svařovací stanice. Řídicí systém robota vlastní také svařovací stanice a dochází ke svařování. Stejně jako externí osu *Robot ext axis 2*. Dveře na pozici 1 jsou otevřené. Otočný stůl vlastní blok pro otáčení stolu *Rotation logistic* a čeká na příkazy. Startovací tlačítko cyklu čeká na stisknutí. Skener kódu zpracovává načtený kód.

Při případu restartu buňky došlo k události *bMessage* a nějaký objekt si vyžádal restart *bResetNecessary* a tak je buňka ve stavu *bReset*. Schránky *jigů*, roboti, dveře, otočný stůl mají v rozkazu provést inicializaci.

4 VYHODNOCENÍ NAVRŽENÝCH A REALIZOVANÝCH ZMĚN

Původní program obsahuje celou logiku řízení svařovací buňky včetně možnosti zobrazování bezpečnostních událostí. Tyto události slouží pro operátora svařovací stanice, aby informovaly o chybách a případně neumožnily pokračování ve výrobě. Při odeznění, či vyřešení problému událost zmizí. Pokud se událost objevuje náhodně nebo v nějakém intervalu na krátkou dobu, servisní technik neví, při jakých činnostech zařízení docházelo k chybě. Při řešení problému tak musí simulovat podmínky, při kterých k chybě došlo, aby přišel na jeho příčinu a problém odstranil. To tedy vedlo k potřebě události monitorovat a vyhodnocovat.

Vytvořený program byl testováním odladěn, aby mohl být nasazen do provozu. Testování proběhlo formou simulačního spouštění jednotlivých bezpečnostních událostí s tím, že se pracovalo s reálnými hodnotami datových bloků svařovací buňky. Vytvořený program nemohl být v době zpracování diplomové práce zcela ověřen na reálném zařízení z důvodu, že nově postavená a plně funkční svařovací buňka, která se odváží k zákazníkovi je ve stavu, kdy nedochází k bezpečnostním událostem. Ty se projeví časem, až po opotřebením. Ověření nemohlo být provedeno ani na starších buňkách, které jsou v provozu u zákazníka. Je to z důvodu, že tyto zařízení nepodporují nově zavedený standart programování firmy. Podle plánu je tedy sběr dat a případná úprava programu odhadována přibližně nejdříve za tři čtvrtě roku, než se dostanou nové buňky s integrovaným program k zákazníkům a než dojde jejich opotřebením, způsobující bezpečnostní události.

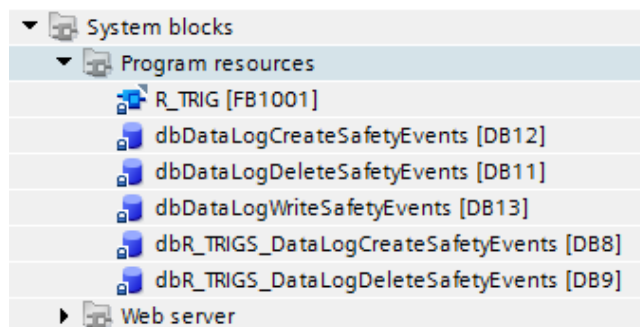
4.1 Integrace do stávajícího programu

1. Nakopírování složky *SafetyLogging* obsahující funkční a datové bloky do složky *Program blocks*.



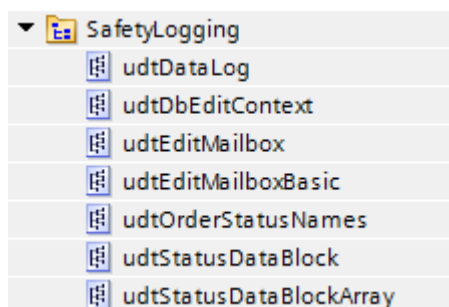
Obrázek 62 – Integrace hlavní program

2. Nakopírování obsahu složky *System blocks/Program resources* do stejné složky původního programu. Složka obsahuje datové bloky systémových proměnných.



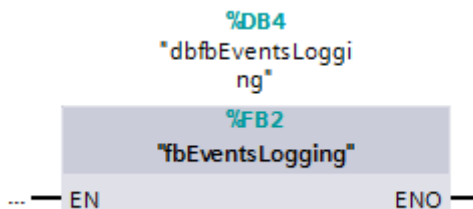
Obrázek 63 – Integrace systémové proměnné

3. Nakopírování složky *SafetyLogging* ze složky *PLC data types* do stejné složky původního programu. Složka obsahuje vytvořené datové struktury.



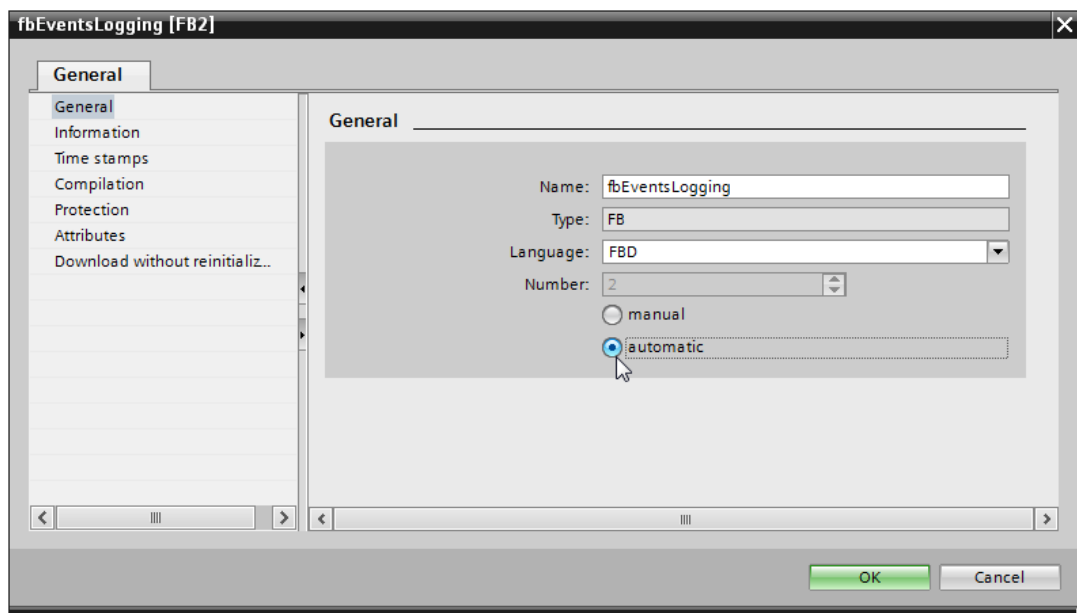
Obrázek 64 – Integrace datové struktury

4. Do hlavního bloku *Main* původního programu vložit *fbEventsLogging*.



Obrázek 65 – Integrace, úprava Main programu

5. Nakopírované funkční bloky se musí re inicializovat. Blokům se přiřadí další volné číslo v programu. Pomocí *Properties, General, Number – Automatic*.



Obrázek 66 – Integrace, změna číslování bloků

6. Na síti vedle složky s původní webovou stránkou *Web2PLCInitial* bude umístěna nová složka *Web2PLCInitial WithSafetyLogs*. Dále se složka vybere v PLC jako *User-defined Web Page*.
7. Pro procházení CSV souboru je potřeba změnit oddělovač seznamu a desetinný oddělovač. Windows -> Ovládací panely -> Oblast a jazyk -> Další nastavení-> Oddělovač seznamu', Desetinný oddělovač ';

Návod jak integrovat vytvořený program a jak jej používat byl vytvořen i pro firmu. Tento návod je dodáván k práci jako příloha.

4.2 Možná rozšíření

Po důkladném testování v provozu by se tato práce mohla rozšířit o následující body:

- CSV soubory by se automaticky odesílaly na uložení firmy po překročení limitu záznamů, nedocházelo by tedy ke ztrátě dat.
- Na uložení by docházelo k třídění souborů podle zákazníků a označení svařovacích buněk.
- Záznamy by se poté mohly vyhodnocovat. Sledovala by se četnost jednotlivých událostí. Kontingenční tabulkou by se zpracovávala závislost jednotlivých událostí a stavů buněk, při kterých vznikají. Podrobnější náhled na problém by umožnila kontingenční tabulka závislosti událostí a příkazů pro jednotlivé schránky.

Například při jaké specifické činnosti robota, či otočného stolu dochází k chybě. V případě, kde by byla vysoká závislost, tak by došlo k zaměření se na tuto kritickou sekci a byla by snaha o minimalizování chyby.

Stávající stav je takový, že až na malé množství buněk, nepřetržitým přístupem k buňkám firma AWL momentálně nedisponuje. Buňky jsou propojeny pouze na intranet v dané firmě zákazníka. Při nutnosti online přístupu k buňce je uděleno zákazníkem firmě AWL jednorázový přístup.

ZÁVĚR

Podle požadavků bylo provedeno monitorování bezpečnostních událostí, jímž při jejich aktivaci byl přidělen identifikátor, byl zaznamenán čas způsobených událostí a byla jim udělena priorita v závislosti na povaze události. Ve stejném čase aktivní události byl zaznamenán aktuální stav buňky a stav schránek objektů. Tyto zaznamenaná data popisující informace o jedné události a stavu celé buňky jsou považovány za jeden záznam. Tento záznam byl uložen do pole o zvolené délce 100 záznamů s tím, že se záznamy pro překročení limitu přepisují. Vizualizace záznamů byla zprostředkována pomocí webserveru daného PLC. Tento způsob vizualizace byl zvolen z ohledem na uživatele, který je ve většině případů servisní technik, aby nemusel používat pro procházení záznamů další software.

Pro procházení všech záznamů najednou je k dispozici CSV soubor, kde jeden řádek prezentuje jeden záznam, sloupce jsou poté jednotlivé proměnné. Tento soubor lze vytvořit, či smazat z vytvořené webové stránky. Tento web primárně slouží k procházení jednotlivých záznamů jeden po druhém. Uložená data jsou zde rozdělena do sekcí, které obsahují informace o aktivní události, data popisující stav buňky a stavy schránek. Tak rozdělená data slouží k lepší orientaci a analýze vzniklé události. Z dat byla provedena i základní statistika ukazující počet jednotlivých priorit a stáří vzniklých událostí zařazených do časových intervalů. K programu byl napsán plán pro integraci do stávajících programů. Podobný plán byl vytvořen i pro firmu, kde se popisuje jak zacházet s programem a jak jej integrovat.

Vytvořený program je snadno integrovatelný do řídicího programu buňky a rozšiřuje tak jeho funkční možnosti. Výsledek práce by měl mít při nasazení do procesu za následek zkrácení doby pro hledání způsobených chyb, od čehož se odvíjí zkrácení prostojů svařovací buňky ve výrobním procesu.

SEZNAM POUŽITÉ LITERATURY

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. vyd. Praha: BEN technická literatura, 1999, 223 s. ISBN 80-860-5658-9.
- [2] BOLTON, W. *Programmable logic controllers*. 5th ed. Boston: Newnes, 2009, xii, 400 p. ISBN 978-185-6177-511.
- [3] SYSALA, Tomáš. Programovatelné automaty [online prezentace]. 2014 [cit. 2016-02-28]. Dostupné z: <http://czv.wz.cz/prednasky.zip>
- [4] CENDELÍN, CSC., doc. Ing. Jiří. Historie programovatelných automatů a jejich současné efektivní použití. *Automa* [online]. 2003, **2003**(6), 1 [cit. 2016-01-27]. Dostupné z: <http://automa.cz/historie-programovatelnych-automatu-a-jejich-soucasne-efektivni-pouziti-28831.html>
- [5] KOVÁŘ, Ing. Josef a Ing. Ladislav ŠMEJKAL, CSC. *Programování PLC* [online]. 1. Zlín, 2008 [cit. 2016-01-27]. Dostupné z: http://www.spszl.cz/soubory/plc/programovani_plc.pdf
- [6] Siemens v České republice: SIEMENS AG. W5.siemens.com [online]. [cit. 2016-02-28]. Dostupné z: https://w5.siemens.com/web/cz/cz/corporate/portal/home/o_nas/Pages/profil_spolecnosti.aspx
- [7] SIMATIC S7-1500. [Http://w3.siemens.com/](http://w3.siemens.com/) [online]. [cit. 2016-02-28]. Dostupné z: <http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-1500/Pages/default.aspx>
- [8] SIEMENS. *Technical data: 6ES7516-3FN01-0AB0* [online]. In: . s. 1 [cit. 2016-01-27]. Dostupné z: <https://support.industry.siemens.com/cs/pd/578299?pdtd=td&lc=en-WW>
- [9] RAKUŠAN, Ondřej. Nová generace LOGO!8 [online prezentace]. 2014 [cit. 2016-02-28]. Dostupné z: http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/logo/prezentace_logo8_2014_cz.pdf
- [10] KUBÁŇ, Vojtěch. Nová generace bezpečnostních automatů: Simatic S7-1500F. *Automa* [online]. 2014, **2014**(6), 2 [cit. 2016-01-27]. Dostupné z: <http://automa.cz/res/pdf/52430.pdf>
- [11] RADA, Václav. Teorie měření a regulace [online prezentace]. 2012 [cit. 2016-02-28]. Dostupné z: <http://www.fce.vutbr.cz/tst/rada.v/TMaR/t-mar-2011-bezp-%C4%8CSNENISO.ppt>

- [12] SIEMENS. *SIMATIC S7-1500: Getting Started* [online]. 2014, **2014**(5), 126 [cit. 2016-01-27]. Dostupné z: http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/software_complete_en.pdf
- [13] PROFINET. *Siemens* [online]. [cit. 2016-05-04]. Dostupné z: <http://www1.siemens.cz/ad/current/?vw=0&ctxnh=bae95e75f4&ctxp=home&acceptcookies=true>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AC	Alternating Current, Střídavý proud
AWL	Automated Welding, Název firmy zadávající projekt
AWP	Automation Web Rprogramming, Příkazy pro propojení webu a PLC
CPU	Central processing unit, Centrální procesorová jednotka
CSV	Comma-separated values, Hodnoty oddělené čárkami
DB	Data Blocks, Datové bloky
DC	Direct Current, Stejnosměrný proud
DIN	Německý ústav pro průmyslovou normalizaci
DP	Decentralized Peripherals, decentralizované periférie
FB	Function Blocks, Funkční bloky
FBD	Function Block Diagram, Programovací jazyk – Funkční bloky
HMI	Human Machine Interface, Rozhraní člověk-stroj
HTML	Hyper Text Markup Language, Hypertextový značkovací jazyk
IEC	Mezinárodní elektrotechnická komise
IL	Instruction List, Seznam příkazů
IP	Internet Protocol, internetový protokol síťové vrstvy
ISO	Mezinárodní organizace pro normalizaci
I/O	Input/Output, Vstup/Výstup
LAD	Ladder Logic, Programovací jazyk – Kontaktní schéma
LD	Ladder Diagram, Kontaktní schéma
MAC	Media Access Control, identifikátor síťového zařízení
PC	Personal computer, Osobní počítač
PLC	Programmable logical controler Programovatelný automat
PN	PROFINET

PROFIBUS	Process Field Bus, průmyslová sběrnice
PROFINET	Process Field Net, průmyslová sběrnice na základech Ethernetu
RFID	Radio Frequency Identification, Identifikace na rádiové frekvenci
SCL	Structured Control Language, Programovací jazyk – Strukturovaný text
SFC	Sequential Function Chart, Sekvenční funkční graf
ST	Structured Text, Strukturovaný text
STL	Statement List, Programovací jazyk – Seznam příkazů
TCP	Transmission Control Protocol, Transportní vrstva protokolu
TIA	Totally Integrated Automation, Vývojový software
URL	Uniform Resource Locator, Jednotná adresa zdroje

SEZNAM OBRÁZKŮ

Obrázek 1 – Hardwarové schéma PLC [2]	12
Obrázek 2 – Cyklus PLC programu.....	13
Obrázek 3 – CPU 1516F-3 PN/DP	15
Obrázek 4 – Přídavné moduly pro S7-1500.....	17
Obrázek 5 – Ostatní PLC zástupci podle kategorií [9]	18
Obrázek 6 – Výběr zařízení	19
Obrázek 7 – Náhled pracovního prostředí	20
Obrázek 8 – Vizualizace PLC.....	20
Obrázek 9 – Programové bloky	21
Obrázek 10 – Programovací jazyky.....	22
Obrázek 11 – Náhled na webserver	23
Obrázek 12 – Svařovací buňka PN23345	24
Obrázek 13 – Svařovací buňka s bezpečnostními prvky	25
Obrázek 14 – Svařovací buňka s HMI prvky	27
Obrázek 15 – Bezpečnostní dveřní systém	28
Obrázek 16 – Propojení pomocí PROFINET	31
Obrázek 17 – Propojení pomocí 24V vodičů.....	32
Obrázek 18 – Propojení pomocí 24V bezpečnostních kabelů a pneumatiky	33
Obrázek 19 - Ukázka prolínání programů	35
Obrázek 20 – Datový blok HMI Safety Messages	36
Obrázek 21 – Datový blok Context	39
Obrázek 22 – Datový blok Mailbox	40
Obrázek 23 – Vytvořené bloky	41
Obrázek 24 – Ukázka kódu z <i>fbEventsLogging</i>	42
Obrázek 25 – Datový blok Status Block.....	43
Obrázek 26 – Ukládání informací o události	43
Obrázek 27 – Zapisování Context	44
Obrázek 28 – Struktura OrderStatusName	44
Obrázek 29 – Zapisování jména příkazů a statusů	44
Obrázek 30 – Zapisování jmen názvů schránek a vlastníků.....	45
Obrázek 31 – Zapisování doplňujících dat	45
Obrázek 32 – Uložení záznamu	45

Obrázek 33 – Mazání záznamů.....	46
Obrázek 34 – Obsluha webové schránky, režim 0.....	47
Obrázek 35 – Obsluha webové stránky, režim 0	47
Obrázek 36 – Obsluha webové stránky, režim 2	47
Obrázek 37 – Obsluha webové stránky, režim 3	47
Obrázek 38 – Obsluha webové stránky, režim 4, tlačítka	48
Obrázek 39 – Obsluha webové stránky, režim 4, výběrové menu	49
Obrázek 40 – Spuštění funkce pro statistiku	49
Obrázek 41 – Nulování statistických hodnot.....	49
Obrázek 42 – Statistické vyhodnocení záznamů	50
Obrázek 43 – Tvorba hlavičky.....	50
Obrázek 44 – Vytvoření csv souboru, hlavní funkce.....	51
Obrázek 45 – Vytvoření csv souboru, pomocné proměnné.....	52
Obrázek 46 – Podmínky a logika pro zápis dat do souboru	53
Obrázek 47 – Ukázka zápisu dat do Data	53
Obrázek 48 – Funkce pro zapsání dat do souboru	53
Obrázek 49 – Zapisování dat, pomocné proměnné.....	54
Obrázek 50 – Funkce pro smazání csv souboru	54
Obrázek 51 – Pomocné proměnné funkce pro mazání csv souboru	55
Obrázek 52 – AWP příkazy	55
Obrázek 53 – Čtení dat z PLC	56
Obrázek 54 – Provázání tlačítek a proměnných	56
Obrázek 55 – CSV soubor na webserveru	57
Obrázek 56 – Ukázka csv souboru	57
Obrázek 57 – Původní webová stránka.....	58
Obrázek 58 – Webová stránka, horní část	59
Obrázek 59 – Webová stránka, dolní část.....	60
Obrázek 60 – Ukázka dat při automatickém režimu.....	61
Obrázek 61 – Ukázka dat při restartu	62
Obrázek 62 – Integrace hlavní program	64
Obrázek 63 – Integrace systémové proměnné	65
Obrázek 64 – Integrace datové struktury.....	65
Obrázek 65 – Integrace, úprava Main programu	65
Obrázek 66 – Integrace, změna číslování bloků	66

SEZNAM TABULEK

Tabulka 1 – Sekce svařovací buňky	24
Tabulka 2 – Povolení přístupu	26
Tabulka 3 – Název a popis zařízení	28
Tabulka 4 – Přehled událostí	36
Tabulka 5 – Seznam schránek	40

SEZNAM PŘÍLOH

PŘÍLOHA P I: MANUÁL K VYTVOŘENÉMU PROGRAMU PRO FIRMU AWL

PŘÍLOHA P II: CD-ROM