

Aplikace zpracování dat pro automatický gravírovací stroj s kolaborativním robotem

Aleksandr Rozhnov

Bakalářská práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Aleksandr Rozhnov**
Osobní číslo: **A16738**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **prezenční**

Téma práce: **Aplikace zpracování dat pro automatický gravírovací stroj s kolaborativním robotem**

Téma anglicky: **A Data Processing Application for an Automatic Engraving Machine with a Collaborative Robot**

Zásady pro vypracování:

1. Navažte komunikaci mezi PLC a počítačem pomocí TCP/IP protokolu, včetně možnosti čtení a zápisu dat do paměti PLC.
2. Navrhněte způsob zpracování dat ze souborů s příponou "*.txt".
3. Navrhněte algoritmus pro posílání zpracovaných dat do PLC a zajistěte kontrolu jejich správnosti. Poslaná data a data z paměti PLC by měla být stejná.
4. Zajistěte monitorování stavu tištění štítků v reálném čase.
5. Navrhněte vhodné uživatelské prostředí pro práci s programem, včetně jeho odolnosti proti neúmyslným chybám uživatele.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Praha: Vydavatelství ČVUT, 1998. ISBN 80-010-1766-4.**
2. **MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty II. Praha: Vydavatelství ČVUT, 2000. ISBN 80-010-2096-7.**
3. **ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.**
4. **NAGEL, Christian. Professional C# 7 and .NET Core 2.0. 7-th edition. Indianapolis: Wrox, 2018. ISBN 978-1119449270.**
5. **SINGLETON, James. ASP.NET Core 1.0 High Performance. 1-st edition. Birmingham: Packt Publishing, 2016. ISBN 978-1785881893.**
6. **http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/programming-guideline-for-s71200-s71500_2014-09_en.pdf**

Vedoucí bakalářské práce:

doc. RNDr. Ing. Zdeněk Úředníček, CSc.
Ústav automatizace a řídicí techniky

Konzultant:

Bc. Tomáš Beck

Datum zadání bakalářské práce:

21. prosince 2018

Termín odevzdání bakalářské práce:

15. května 2019

Ve Zlíně dne 21. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně, dne

Aleksandr Rozhnov v.r.
podpis diplomanta

ABSTRAKT

Tématem dané práce je návrh softwarového řešení zpracování dat v textové podobě pro jejich další možnost nahrání do PLC. V teoretické části této práci jsou popsány některé druhy automatizačních prostředků, používaných v průmyslu, principy zpracování a přenosu dat v TCP/IP sítích a základní pojmy prostředí .NET. Praktická část popisuje principy fungování programu, vytvořeného na základě požadavků společnosti a operátorů stroje. Gravírovací stroj, se kterým pracuje program, byl popsán taky.

Klíčová slova: PLC, S7 Protocol, Snap7, C#, WPF, .NET Framework, Mikro úderové gravírování

ABSTRACT

The subject of this thesis is a suggestion of a software solution, used for text data processing for the possibility of its further sending to the PLC. Theoretical part of this thesis describes some of the machines used in industrial production, principles of processing and sending the data through TCP/IP networks and basic features of .NET platform. Practical part describes the way the program works, which was based on firm's and its operators' demands. The engraving machine, which program works with, was also described.

Keywords: PLC, S7 Protocol, Snap7, C#, WPF, .NET Framework, Dot-Peen Engraving

Tímto bych chtěl poděkovat panu doc. RNDr. Ing. Zdeňku Úředníčkovi, CSc. za cenné poznámky, trpělivost a celkovou profesionalitu při vedení této bakalářské práce.

Dále bych chtěl poděkovat firmě AWL-Techniek CZ s.r.o., za poskytnutí možností získat pracovní zkušenosti nejenom v rámci tvorby dané práce a i obecně. Poděkování taky patří Bc. Tomášovi Beckovi za trpělivost, kladný přístup a ochotu pomoci.

Nejvíce bych chtěl poděkovat mé rodině a kamarádům, které mě celou dobu podporovali a pomáhali.

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

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 AUTOMATIZAČNÍ PROSTŘEDKY.....	11
1.1 MIKRO ÚDEROVÉ GRAVÍROVACÍ SYSTÉMY	11
1.2 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT.....	11
1.2.1 Historický vývoj PLC	12
1.2.2 Hlavní výhody a nevýhody PLC	13
1.2.3 Struktura PLC.....	13
1.2.4 Programování PLC.....	14
1.3 PRŮMYSLOVÉ ROBOTY A JEJÍCH APLIKACE	15
1.3.1 Historický vývoj robotů	15
1.3.2 Součásti průmyslových robotů.....	15
1.3.3 Využití robotů při automatizaci výrobních procesů.....	17
2 METODY KOMUNIKACE MEZI PLC A PC	18
2.1 TCP/IP MODEL POČÍTAČOVÝCH SÍTÍ.....	18
2.2 PŘENOS DAT MEZI PLC FIRMY SIEMENS A POČÍTAČEM	19
2.2.1 Ethernet	19
2.2.2 Internet Protocol.....	22
2.2.3 Transmission Control Protocol	23
2.2.4 ISO-on-TCP	23
2.2.5 S7 Protocol.....	24
3 PROGRAMOVACÍ JAZYK C# A SOFTWAREOVÁ PLATFORMA .NET FRAMEWORK	25
3.1 COMMON LANGUAGE INFRASTRUCTURE.....	25
3.2 PLATFORMA .NET FRAMEWORK.....	25
3.2.1 Common Language Runtime	25
3.2.2 Framework Class Library.....	26
3.2.3 Windows Presentation Foundation	27
3.3 KRÁTKÝ POPIS VLASTNOSTÍ PROGRAMOVACÍHO JAZYKU C#.....	27
II PRAKTICKÁ ČÁST	28
4 POPIS GRAVÍROVACÍHO STROJE.....	29
4.1 POPIS POUŽÍVANÉHO PLC	30
4.2 POPIS SYSTÉMU ZNAČENÍ FIRMY SIC MARKING.....	30
4.3 POPIS POUŽÍVANÉHO ROBOTA	31
4.4 DALŠÍ ČÁSTÍ STROJE	31
4.4.1 PC.....	31
4.4.2 Podavač štítků	31
4.4.3 Rotační stůl.....	32
4.4.4 Plato na štítky.....	33
4.4.5 Vzhled štítku	33

5	ANALÝZA POŽADAVKŮ SPOLEČNOSTI AWL-TECHNIEK CZ S.R.O.	34
5.1	POŽADAVKY FIRMY	35
5.2	POŽADAVKY OPERÁTORŮ	35
6	UŽIVATELSKÉ ROZHRANÍ PROGRAMU PRO PRÁCI S GRAVÍROVACÍM STROJEM	37
6.1	VOLBA JAZYKA ROZHRANÍ	37
6.2	HLAVNÍ OKNO	37
6.3	NASTAVENÍ PŘIPOJENÍ	38
6.4	NASTAVENÍ STRUKTURY DATABÁZE V PLC	38
6.5	NÁHLED DAT ZE ZPRACOVANÉHO SOUBORU	39
6.6	NÁHLED DAT V PAMĚTI PLC	41
7	VLASTNOSTI PROGRAMU PRO PRÁCI S GRAVÍROVACÍM STROJEM	42
7.1	POPIS NÁSTROJE SNAP7	42
7.1.1	Snap7 Client	42
7.1.2	Snap7 Server	42
7.1.3	Snap7 Partner	43
7.2	POPIS ZÁKLADNÍCH FUNKCÍ PROGRAMU „DATA TO PLC PARSER“	43
7.3	DETEKCE CHYB ZA BĚHU PROGRAMU	45
7.3.1	Neúspěšné připojení k PLC	45
7.3.2	Chybná struktura databáze v PLC	46
7.3.3	Chyby při zpracování vstupních souborů	47
7.3.4	Chyby přenášení dat	47
7.3.5	Odolnost programu vůči jeho nesprávnému použití	48
	ZÁVĚR	49
	SEZNAM POUŽITÉ LITERATURY	50
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	53
	SEZNAM OBRÁZKŮ	55
	SEZNAM PŘÍLOH	56

ÚVOD

Od začátku své existenci lidstvo vytváří různé výrobky a pomůcky z důvodu usnadnění života člověka nebo uspokojení jeho potřeb. Poptávka po podobných nástrojích s časem nepřetržitě roste, což vyvolává růst objemu výroby. Z tohoto důvodu člověk je nucen zabývat se optimalizací výrobních procesů nebo automatizovat některé jeho etapy, aby byl schopen splnit poptávky a poskytnout stejnou nebo lepší úroveň kvality výrobků.

Cílem této bakalářské práce je optimalizace automatického procesu gravírování textů na kovových štítcích, přesněji řečeno, urychlení procesu zpracování dat, vstupujících do gravírovacího stroje. Jednou z podmínek, aby stroj mohl začít proces gravírování, je zadat určitá data: gravírovací texty pro každý štítek a počet kopií každého štítku. Podobnou činnost lze vykonat i ručně, ale v tomto případě množství času, potřebného na dokončení tohoto úkolu, bude nesrovnatelně větší, než kdyby ten proces byl proveden pomocí počítačového programu. Kromě toho, při opakovaném ručním konání práce stejného typu roste pravděpodobnost chyby.

Jedním z výstupů této bakalářské práce je počítačový program, který bude schopen zpracovat vstupní soubory s daty pro gravírování, uspořádat tato data do určité struktury a přeposlát je do paměti PLC, který řídí gravírovací stroj. Uživatel jen potřebuje určit v programu soubor, který by chtěl zpracovat a ostatní činnost už by měl udělat program. Takovým způsobem lze urychlit proces gravírování štítků a ušetřit čas operátora stroje, během kterého bude schopen zabývat se jinými problémy výrobních procesů.

I. TEORETICKÁ ČÁST

1 AUTOMATIZAČNÍ PROSTŘEDKY

1.1 Mikro úderové gravírovací systémy

V současnosti existuje spousta způsobů automatického gravírování různých materiálů. Jedním z nich je mikro-úderové gravírování – proces vykreslení obrazu na povrchu výrobku pomocí úderů hrotu, vyrobeného z tvrdé kovové slitiny. Elektromechanické stroje hýbou hrotem pomocí elektromagnetického pole: Na jedné straně hrotu je pevně připevněný permanentní magnet, který je vložen i s hrotem do cívky. Z druhé strany magnet je připevněn k základně pomocí pružiny. Při průchodu proudového impulzu cívkou vytvořený impulz magnetického pole uvede magnet do pohybu směrem dolů. Spolu s magnetem se pohybuje i hrot, který v důsledku tohoto pochybu udělá na povrchu výrobku malý důlek. Jakmile cívkou přestane téct proud, magnet se vrátí do výchozí polohy. Souhrn všech důlků představuje celkový obraz [1].

Daná metoda se používá při značení různých výrobků z plastů nebo kovů. Její zásadní výhody využití jsou: Vysoká odolnost proti opotřebení, nízká závislost výsledků na stavu povrchu výrobku, vysoká rychlost zakreslení obrazu, nenáročné požadavky na prostředí, ve kterém stroj pracuje a hlavně, při gravírování nejsou potřeba žádné spotřební materiály. Obrazem získaným gravírováním může být symbol, písmeno, číslo, čárový nebo jiný druh kódu anebo obrázek. Příkladem obrázku může být firemní značka. Získaný obraz je dost kvalitní, aby ho bylo možné použít při automatickém zpracování obrazu, například pro identifikaci výrobku čtečkou nebo kamerou. Vzhledem k tomu, že moderní gravírovací stroje mohou být ovládané prostřednictvím počítače nebo PLC, existuje možnost vytvoření a uložení velkého množství obrázků a gravírovacích schémat do paměti ovládací jednotky a ty pak mohou být vybrány a využity na základě určitých podmínek při zařazení podobného stroje do výrobní linky z důvodu automatizace výroby [2].

1.2 Programovatelný Logický Automat

Programovatelný Logický Automat (PLC) – je zařízení, určené pro automatické řízení technických a průmyslových procesů. Typickými úlohami pro PLC mohou být jak logické řízení procesů, ve kterém PLC nahrazuje reléovou logiku, tak i spousta dalších typů úloh, například regulace, monitorování procesů nebo měření analogových veličin. Hlavním důvodem použití PLC při automatizaci procesů je jejich spolehlivost, autonomnost a odolnost

vůči rušení, včetně možnosti práce za nepříznivých podmínek. I přes to, že poruchovost PLC je zanedbatelná, při navrhování a obsluze systémů je stále potřeba uvažovat možnost jejich přehřátí, zahlcení atd. Systém může selhávat i pod vlivem “lidského faktoru” [3].

Navíc, v současnosti je kladený důraz na požadavek na minimalizaci přítomnosti operátorů při výrobě. Z toho důvodu řídicí systém musí být schopen rozpoznat a vhodně reagovat na možnou chybu nebo poruchu, která může vzniknout během výrobního procesu. Příkladem podobných poruch mohou být: Mechanické poškození systému nebo jeho části, přehřátí a hrozící požár, únik plynu nebo kapalin atd. Včasné odhalení, diagnostika a řešení podobných problémů jsou možné při použití PLC [3].

Podobné úlohy jsou řešitelné i při pomoci osobního počítače (PC), ale jeho použití v přímém řízení technických procesů je dost riskantní a diskutabilní. Míra spolehlivosti PC je dostatečná pro jeho použití v domácích, kancelářských a laboratorních prostředích, ale ne v technických. PC je příliš citlivý na téměř jakýkoliv druh rušení, prašnost prostředí, obvykle není schopen nepřetržitě pracovat dlouhou dobu a při tom ještě může spotřebovávat velké množství energie. Přesto PC může být použitelný při monitoringu a vizualizaci výrobních procesů, shromáždění statistických údajů i při kontrole přítomnosti a zásahu obsluhujících operátorů [3].

1.2.1 Historický vývoj PLC

Jeden z prvních prototypů PLC s názvem *Modicon 084* byl navržen v roce 1969 firmou Modicon. Jejím zakladatelem byl Dick Morley, který ještě před rokem pracoval ve společnosti General Motors. Tým inženýrů GM, součástí kterého byl pan Morley, dostal za úkol vytvořit programovatelný systém jako náhradu řídicích systémů, pracujících na principu reléové logiky. Nový systém by měl být snadno programovatelný a schopný pracovat v průmyslovém prostředí [4].

První modely PLC byli schopni pracovat s V/V signály, časovači a počítadly. S časem se přidávali další funkce jako práce s analogovými V/V a čísly s desetinnou čárkou. Použití PLC v průmyslu se prudce zvětšilo, když byla implementovaná možnost použití PLC jako PID regulátoru. V současnosti typický PLC je schopen nejenom vykonávat funkce, uvedené výše, ale i navázat komunikaci s různými druhy zařízení nebo sloužit jako WEB-server [4].

1.2.2 Hlavní výhody a nevýhody PLC

Výhody:

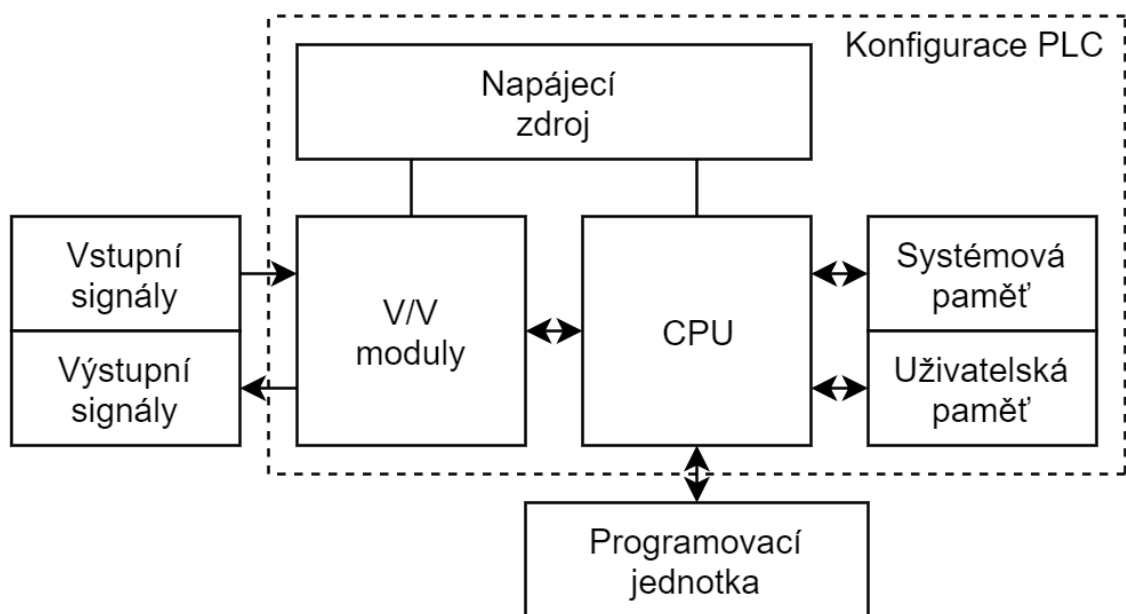
- **Rychlá realizace.** Stačí navrhnout vhodnou konfiguraci PLC, která bude odpovídat požadavkům daného projektu, napsat uživatelský program a připojit PLC na tento systém.
- **Spolehlivost.** PLC jsou navrženy tak, aby mohly spolehlivě pracovat za drsných pracovních podmínek a mají několik úrovní ochran.
- **Pružnost při návržení a provozu.** PLC poskytují možnost doplnit nebo úplně přepsat logiku programu za jakéhokoliv stavu projektu. Časové a materiálové ztráty jsou minimální, na rozdíl od reléových systémů, kde podobné změny mohou být velice náročné nebo neproveditelné.
- **Možnosti komunikace.** PLC jsou schopny komunikovat a přenášet informaci jak do systémů v podřízené úrovni (oblast senzorů, akčních členů atd.), tak i s jinými PLC nebo zařízeními, které mohou být umístěny v nadřízené úrovni.

Nevýhody:

- **Zpožděná odezva.** Na rozdíl od integrovaných schémat, PLC mají větší dobu odezvy. Tato doba je většinou závislá na obtížnosti uživatelského programu a rychlosti procesoru v PLC. Doba odezvy se obvykle pohybuje v rozmezích jednotek až desítek milisekund – bohatě postačující pro většinu aplikací v průmyslu.
- **Diskretizace.** Programy v PLC jsou vykonány v cyklech. Na začátku cyklu hodnoty na vstupech do PLC jsou zkopírovány do paměti PLC a pak program pracuje jen s těmi hodnotami, které má v paměti. Kvůli tomu PLC není schopno okamžitě reagovat na změny na vstupech, pouze začne-li nový cyklus.
- **Posloupnost zpracování.** Uživatelský program se vykonává v tom pořadí, ve kterém byl napsán. Na tuto skutečnost je potřeba dávat pozor při projektování a programování řízení systémů se zpětnými vazbami [3].

1.2.3 Struktura PLC

Přesto, že různé modely PLC mohou být od sebe docela odlišné z hlediska jejich složení, jejich obecnou strukturu lze popsat pomocí následujícího schématu:



Obr. 1. Obecná struktura PLC

- **CPU** – hlavní součást PLC, vykonávající logické funkce a programy. CPU zpracovává vstupy získané od V/V modulů pomocí uživatelského programu a na základě výsledků nastavuje hodnoty výstupních signálů prostřednictvím V/V modulů [5].
- **Systémová paměť** – oblast paměti PLC, obsahující obrazy vstupů a výstupů a systémové programy. V této části paměti je taky umístěná operační paměť.
- **Uživatelská paměť** – obsahuje uživatelské programy a data [3].
- **V/V moduly** – jeden nebo více modulů, pomocí kterých probíhá sbírání informací a případné řízení strojů.
- **Programovací jednotka** – externí zařízení pomocí kterého lze programovat PLC, případně provádět jeho diagnostiku. V roli programovací jednotky lze použít PC se speciálním programem (např. TIA Portal, Mosaic) [5].

1.2.4 Programování PLC

V současnosti PLC podporují několik docela odlišných typů programovacích jazyků. Každý typ má svoje vlastností a rozsahy použití:

- **Jazyky mnemokodů** (Statement List, STL) – jazyky podobné assembleru na PC. Jsou strojově orientované – každému příkazu jazyka odpovídá instrukce procesoru

PLC. Používají se, když je potřeba zajistit úplnou kontrolu běhu programu nebo co nejlépe jej přizpůsobit pod určitou model PLC.

- **Jazyky reléových schémat** (Ladder Diagram, LAD) – grafické jazyky programování, ve kterých programy jsou zobrazené ve formě reléových schémat. Jsou vhodné pro návrh jednoduchých programů a jsou snadno použitelné i pro lidi, které neumí programovat tradičními metody.
- **Jazyky logických schémat** (FBD) – další typ grafických programovacích jazyků. Programování je založeno na stejném principu jako u LAD jazyků. Rozdíl je v tom, že za programovací prvky se používají jak základní logické funkce, tak i ucelené funkční bloky, např. čítače, časovače nebo různé logické instrukce [3].
- **Vysokourovňové jazyky** – programovací jazyky, mající syntaxi podobnou počítačovým jazykům C nebo Pascal. Používají se, pokud je potřeba napsat rozsáhlý program. Mohou podporovat principy objektově-orientovaného programování.

1.3 Průmyslové roboty a jejich aplikace

Průmyslový robot – řízený, programovatelný mechanismus, schopný vykonávat pohyby ve třech nebo více osách. Je důležitou součástí průmyslové automatizace. [6]

1.3.1 Historický vývoj robotů

V roce 1934 pan Willard Pollard Jr. podal patentovou žádost na stříkací stroj určený k nanášení nátěru. Daný stroj byl složen z mechanického manipulátoru a elektrického řídicího systému, což znamená, že ve své podstatě to byl návrh prvního průmyslového robota. Ovšem prvním sériově vyráběným robotem byl Unimate – jeho první prototyp byl zařazen do montážní linky společnosti General Motors v roce 1961. S časem rychlost vyvíjení různých průmyslových robotů rostla a k roku 1975 už dosahovala 30 % ročně. Ten proces stále pokračuje i dnes aby bylo možné zajistit splnění rostoucích požadavků firem, používajících roboty při výrobě svých produktů [7][8].

1.3.2 Součásti průmyslových robotů

Robotický systém lze popsat jako mechanismus, jehož součástmi jsou *manipulátor, zápěstí, koncový efektor, akční členy, senzory a kontrolér*.

Manipulátor – mechanismus, složený ze článků, kloubů a dalších elementů. Manipulátor lze považovat za robota, pokud ten manipulátor má na jednom z jeho konců zápěstí s koncovým efektozem a celý ten mechanismus je řízený řídicím systémem.

Každý robot má tzv. *pracovní prostor* – celkový prostor, kterého ten robot je schopen dosáhnout svým koncovým efektozem.

Článek je tuhé těleso, které je schopné pohybovat a měnit svoji orientaci vzhledem k ostatním článkům.

Kloub – místo spojení dvou článků. Existuje 2 typy kloubů: rotační (**R**) a lineární (**P**), každý z kterých definuje jakým způsobem články budou měnit svoji polohu a orientaci vůči sobě. Každý kloub má osu, okolo které bude daný typ pohybu vykonán. Na základě vzájemné polohy os dvou sousedních kloubů lze odvodit 3 druhy uspořádání: paralelní (**||**), ortogonální (**⊥**) a perpendikulární (**⊥**). Ze třech článků, spojených mezi sebou dvěma klouby, lze vytvořit 72 možné konfigurací manipulátoru:

$$a^b * c^d = 2^3 * 3^2 = 72 \quad (1)$$

kde a – počet typů kloubů, b – počet článků, c – počet druhů uspořádání kloubů, d – počet kloubů.

Zápěstí – kombinace z několika kloubů, které jsou umístěné mezi koncovým efektozem a posledním článkem před nim. Nejčastějším druhem používaného zápěstí je sférické. Tento typ zápěstí je složený ze třech rotačních kloubů, jejichž osy se protínají ve stejném bodě, který se nazývá *bod zápěstí*.

Koncový efektor – část robota, která je připojená k jeho zápěstí. Pomoci efektoru robot je schopen vykonávat ten druh práci, ke kterému on je určený.

Akční členy – hnací prvky, umístěné v kloubech manipulátoru, úkolem kterých je poskytování potřebného výkonu pro změnu polohy článků nebo odporu vůči různým silám, působícím na manipulátor. Důležitou podmínkou akčního členu je možnost jeho řízení. Pokud ve kloubu je umístěn akční člen, tento kloub se považuje za *aktivní*. *Pasivní* klouby nemají akční členy.

Senzory jsou prvky, umístěné v různých částech robota, poskytující informací jak o stavu součástí robota, tak i ob okolí, působícím na robota. Získané informace se posílají do kontroléru, který na základě vyhodnocení těchto informací určuje okamžitý stav robota.

Kontrolér – zařízení, složené z procesoru a softwaru. Kontrolér sbírá a analyzuje informaci získanou od senzorů, určuje pohyb jednotlivých částí robota a poskytuje interakci mezi robotem a dalšími zařízení [9].

Ve většině případů ke kontroléru lze připojit *Teach Pendant* – zařízení, které poskytuje uživatelské rozhraní mezi robotem a operátorem. S jeho pomocí lze manuálně řídit robota, vytvářet nebo měnit programy a získávat informaci ob okamžitém stavu robota.

1.3.3 Využití robotů při automatizaci výrobních procesů

Průmyslové roboty se používají k plnění různých typů práce a většinou nahrazují činnost člověka. Důvodem k tomu může být práce v nebezpečném pro člověka prostředí, potřeba vykonávat opakující druhy práce, které mohou vyžadovat vysokou míru pozornosti a/nebo přesnosti během dlouhého času. Příklady takových typů prací mohou být svařování, manipulace s předměty, rozstříkání barev, montáž součástek atd. Ve většině případů náhrada člověka robotem u určitých typů práce je pro firmu prospěšná.

2 METODY KOMUNIKACE MEZI PLC A PC

2.1 TCP/IP model počítačových sítí

TCP/IP model je síťový model, popisující způsoby přenosu dat od adresantu k adresátovi. Je popsán standardem RFC 1121. Je to sada protokolů, které popisují, jakými způsoby probíhají paketizace, zpracování, přenos, směrování a příjem dat.

Každý protokol patří do jedné ze 4 vrstev modelu:

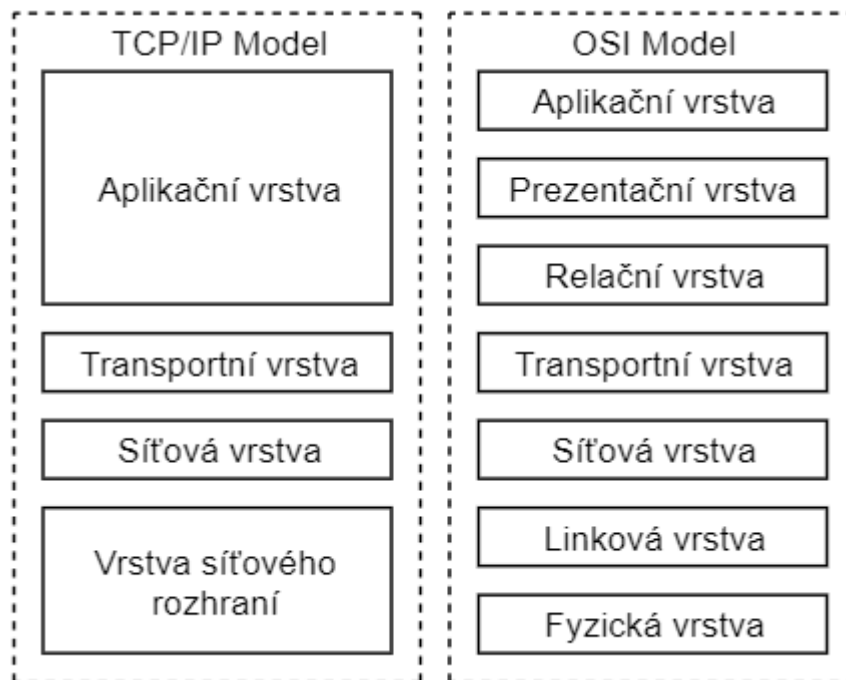
Vrstva síťového rozhraní – první, nejnižší vrstva modelu. Popisuje principy fyzického přenosu dat nebo, řečeno jinak, způsoby přenosu bitů informace v poskytovaném mediu (např. kroucená dvojlinka nebo optické vlákno). Příklady protokolů: *Ethernet*, *Token Ring*, *Fiber Distributed Data Interface*.

Síťová vrstva – druhá vrstva modelu, jejíž úkolem je ukládání dat do speciálních balíčků – *IP Diagrams*. Dané balíčky obsahují informaci, která je používána při přenosu dat mezi různými sítěmi. Hlavním úkolem této vrstvy je poskytování možnosti pro hosta posílat a dostávat informaci přes síť. Příklady protokolů: *Internet Protocol*, *Internet Control Message Protocol*, *Address Resolution Protocol* atd.

Transportní vrstva – třetí vrstva modelu. Poskytuje přístup do sítí pro aplikace a kontrolu správnosti posílaných dat. Navíc k tomu, rozděluje datové balíčky mezi jednotlivými aplikacemi a určuje pro kterou aplikaci je určen posílaný balíček pomocí speciálního čísla – *portu*. Příklady protokolů: *Transmission Control Protocol* a *User Datagram Protocol*.

Aplikační vrstva – čtvrtá, nejvyšší vrstva modelu, na které pracují většina síťových aplikací. Data pro výměnu mezi různými aplikacemi v síti se vytváří v této vrstvě. Příklady protokolů: *HyperText Transfer Protocol*, *File Transfer Protocol*, *Simple Mail Transfer Protocol* atd. [10].

Existuje další síťový model – ISO/OSI. Na rozdíl od modelu TCP/IP, má 7 vrstev a ve své podstatě představuje konceptuální model určený pro lepší popis jednotlivých částí sítě. Model TCP/IP je starší než ISO/OSI a byl vytvořen na základě již existujících protokolů, které byli používány při založení sítě Internet. TCP/IP má praktický význam – důvodem vytvoření tohoto modelu bylo řešení problémů [11].



Obr. 2. Struktury síťových modelů TCP/IP a OSI

2.2 Přenos dat mezi PLC firmy SIEMENS a počítačem

PLC z modelového řádu S7-1500 společnosti Siemens jsou schopní komunikovat s dalšími zařízeními pomocí několika protokolů, založených na standardu Industrial Ethernet (PROFIBUS, PROFINET, MPI atd.). IE je interpretací protokolu Ethernet v průmyslovém prostředí, které požaduje odolnější kabeláž a konektory a taky více zřetelné směrování balíčků s daty z hlediska časování a adresování. Většina protokolů IE modifikují standardní UDP/IP a TCP/IP protokoly přidáním funkcí synchronizace a diagnostiky balíčků, metod automatické korekce atd. Rychlost přenosu dat v IE může dosahovat 1 Gbps. Navíc, protokoly, které používají Ethernet, lze používat i v IE [12].

Pro komunikaci mezi PLC společnosti Siemens a osobním počítačem lze používat standardní Ethernet s protokoly TCP a IP, ovšem pro plynulou komunikaci bude potřeba zprovoznit dělení datového proudu na bloky s daty. Proprietární software pro práci s takovými PLC používá protokol *ISO-on-TCP* (popsaný standardem RCF 1006) jako nadstavbu pro TCP/IP a S7 Protocol na aplikační vrstvě modelu TCP/IP [13].

2.2.1 Ethernet

Ethernet – sada technologií určených ke komunikaci a přenosu balíčků s daty mezi různými zařízeními. Je popsána pomocí standardů skupiny IEEE 802.3.

Lze vyčlenit dva základní elementy Ethernet: *medium pro přenos dat* a *sítové uzly*:

Medium pro přenos dat – popisují způsoby fyzického přenosu signálů v síti. Standard Ethernet umožňuje přenos dat třemi způsoby:

- Kroucená dvojlinka;
- Koaxiální kabel;
- Optické vlákno

Sítové uzly – body ze kterých nebo do kterých se posílají data. Existují 2 typy síťových uzlů:

- **Data Terminal Equipment (DTE)** – jsou zdrojem nebo cílem přenosu dat. Příklady: osobní počítač, server.
- **Data Communications Equipment (DCE)** – zařízení, která dostávají balíčky s daty a posílají je dál přes síť bez změny jejich náplně. Příklady: modem, router, switch, repeater.

Ethernet podporuje několik typů síťových topologií:

- **Point to Point (Bod k Bodu)** – konfigurace, spojující 2 uzly sítě, přičemž tyto uzly mohou být jakéhokoliv typu. Je to nejjednodušší typ konfigurace.
- **Coaxial Bus (Koaxiální sběrnice)** – konfigurace, při které všechny uzly sítě jsou umístěné na jednom koaxiálním kabelu. Maximální počet DTE elementů – 1024, maximální délka kabelu je 500 metrů. V současnosti již skoro nepoužívá.
- **Star network (Hvězdicová topologie)** – tato konfigurace má jeden centrální element, který spojuje spoustu dalších uzlů sítě na základě principu point to point. Je to nejvíc používaná konfigurace sítě současnosti [14].

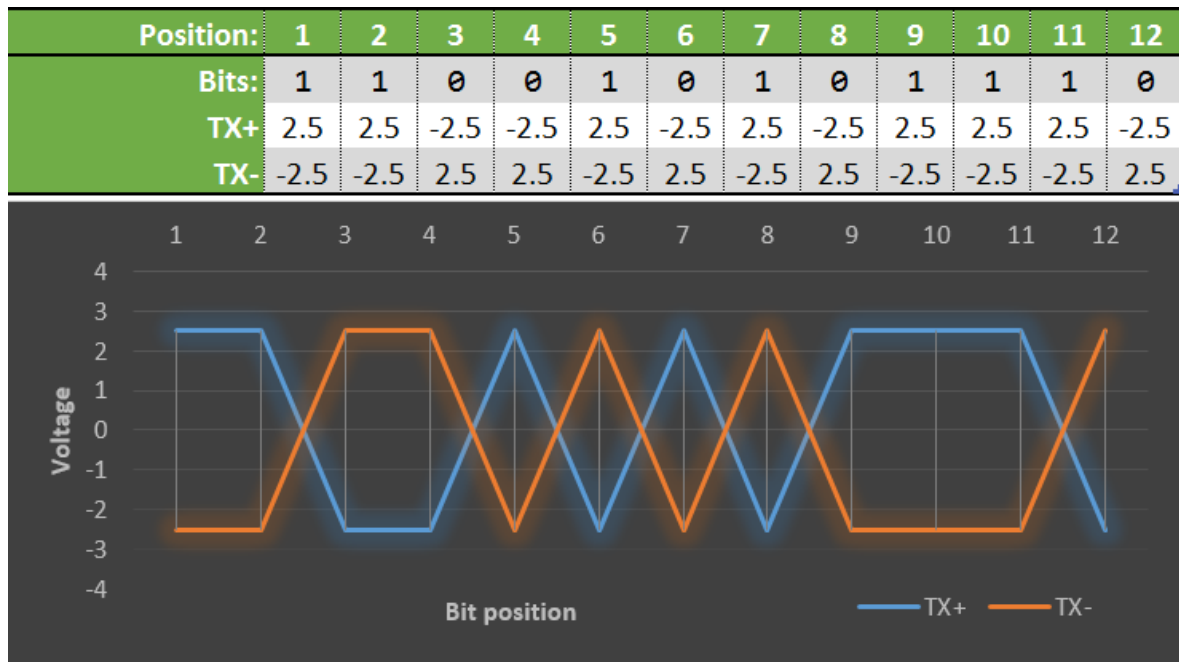
Struktura datového rámce, který se nazývá *Ethernet Frame*, má 5 částí a může mít velikost od 64 do 1518 bajtů v závislosti na množství přenášených dat:

- **Destination MAC-Address** – první část rámce, mající délku 6 bajtů. Obsahuje MAC-adresu adresátu.
- **Source MAC-Address** - druhá část rámce, mající délku 6 bajtů. Obsahuje MAC-adresu adresantu.
- **Ether type** – ukazuje, do které části paměťové schránky je nutné uložit daný balíček. Má délku 2 bajty.

- **Data** – obsahuje data pro vyšší vrstvy, včetně uživatelských dat. Má délku od 46 do 1500 bajtů.
- **Frame Check Sequence** – poslední část rámce, obsahuje checksum předchozích částí rámce. Používá se k detekci chyb datového přenosu a kontrolu celistvosti přijatých dat. Má délku 4 bajty [15].

Prvou (rok 1983) veřejnou realizaci Ethernet byl standard 10Base5, který přenášel data pomocí koaxiálního kabelu délkou do 500 metrů. Teoretická rychlost přenosu byla 10 Mbps, ale ve skutečnosti průměrná rychlost přenosu byla 4 až 6 Mbps. V roce 1995 byl schválen standard IEEE 802.3u, popisující Fast Ethernet, který je masově používán i do dnes. Dodatkem k existujícím vlastnostem byla možnost přenosu dat pomocí kroucené dvojlinky a optického vlákna. Teoretická rychlost přenosu se zvýšila do 100 Mbps [16].

V současnosti se nejvíc používá standard 100BASE-TX. Tato realizace Ethernet používá konektor 8P8C spolu s kroucenou dvojlinkou RJ-45, mající 8 drátů. Data jsou přenášeny pomocí dvou párů drátů: první pár posílá data (TX), druhý je přijímá (RX). Pár, který bude přijímat data nebo posílat, se určuje pomocí Network Interface Card (NIC). NIC, který posílá data přes druhý pár (piny 1 a 2) a přijímá přes třetí pár (piny 3 a 6), se nazývá Media Dependent Interface (MDI). Jeho opakem je Media Dependent Interface Crossover (MDI-X). V osobních počítačích se vždycky používá MDI NIC a proto spojení dvou počítačů bez speciálního kabelu není možné. Je potřeba je spojit přes switch nebo pomocí crossover-kabelu, v opačném případě oba dva počítače budou posílat data přes druhou páru a tím vyvolávat kolize, zatímco třetí pár nebude přenášet žádné data.



Obr. 3. Princip přenosu dat pomocí kroucené dvojlinky [17]

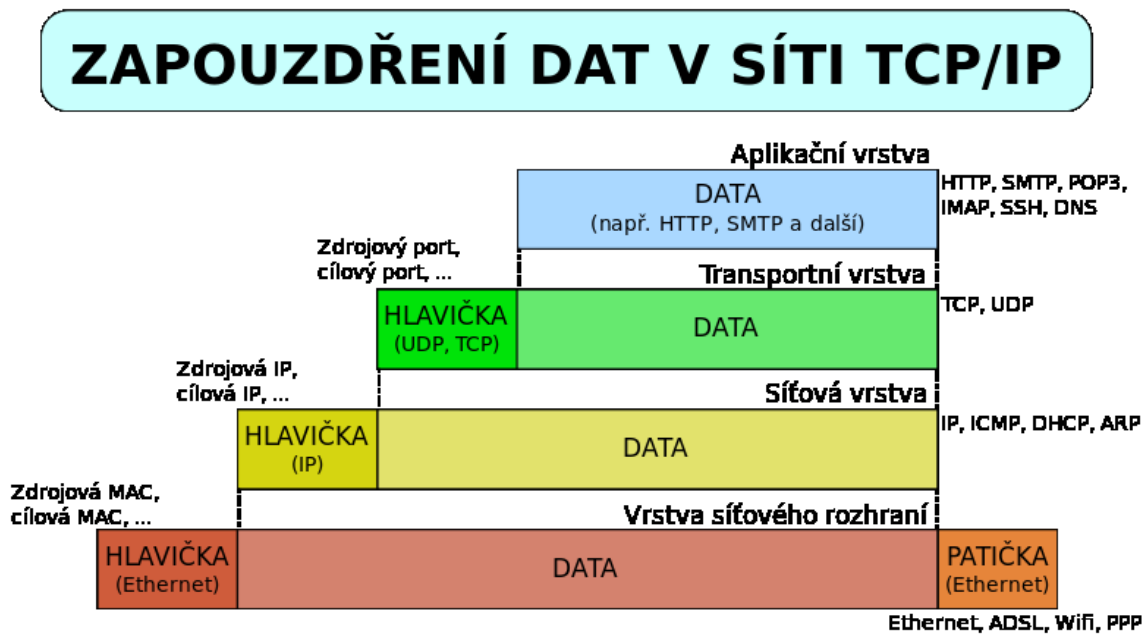
Přenos bitů informace přes TX-pár je možný pomocí rozdělení páru na TX+ a TX- dráty. Při přenosu logické 1 TX+ drát bude mít hodnotu napětí +2,5V, TX- -2,5V. Při přenosu 0, polarity na drátech se obrací: -2,5V na TX+ a +2,5V na TX-. Přijímač kontroluje, který z drátů má větší napětí a na základě tohoto interpretuje získaný bit, což poskytuje ochranu proti rušení [17].

2.2.2 Internet Protocol

Internet Protocol (IP) je protokol popisující metody přenosu dat z jednoho počítače do druhého přes síť Internet. Negarantuje doručení, dodržení pořadí a nepoškozenost dat, protože protokol nevyžaduje navázání propojení mezi odesilatelem a příjemcem. Daný protokol pracuje na síťové vrstvě TCP/IP modelu. Existuje několik verzí IP, ovšem používají se jenom dvě: *IPv4* a *IPv6*. Obecná struktura jejich packetů je stejná – každý packet je složen z dvou částí: *hlavičky* (obsahuje potřebná data k doručení packetu) a *dat* (obsahuje data pro vyšší vrstvy TCP/IP modelu). Podstatný rozdíl mezi IPv4 a IPv6 je v délkách jejich adres: adresa protokolu IPv4 má délku 32 bitů, IPv6 – 128 bitů. To znamená, že podle protokolu IPv4 lze adresovat jenom 4 294 967 296 zařízení, což v současnosti není postačující. Bylo navrženo a implementováno několik způsobů řešení daného problému. Mezi nimi jsou dynamické přidělování adres, využití proxy serverů atd. Nicméně, IPv6 byl vymyšlen jako náhrada IPv4 a očekává se, že s časem bude používán jenom IPv6 [18].

2.2.3 Transmission Control Protocol

Transmission Control Protocol (TCP) je protokol, pracující na transportní vrstvě TCP/IP modelu. Pracuje spolu s IP a zajišťuje navázání propojení jednotlivých aplikací přes síť Internet a spolehlivou výměnu dat ve formě packetového toku mezi nimi [19].

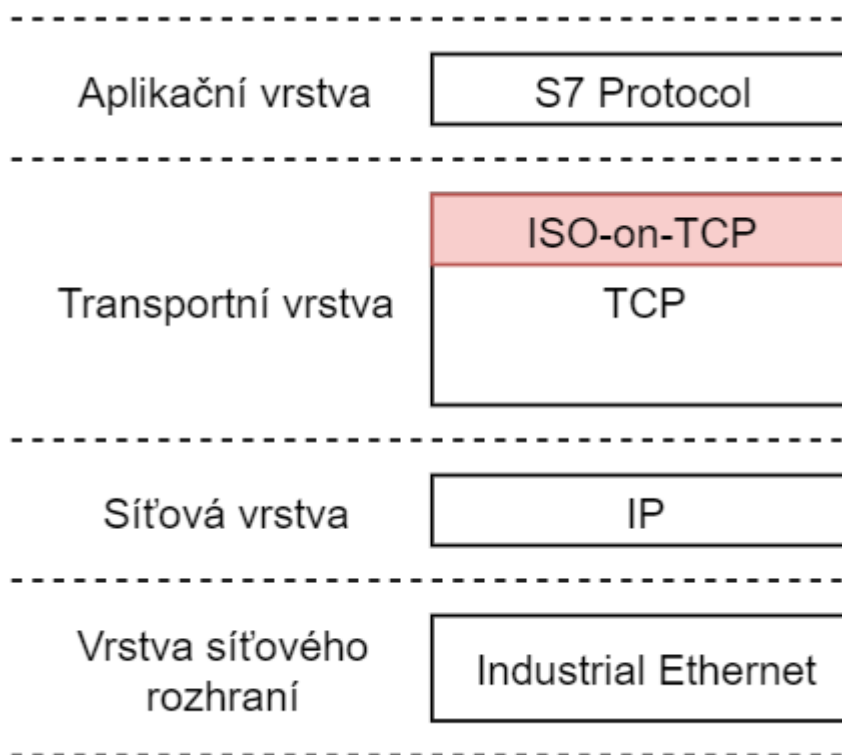


Obr. 4. Zapouzdření dat v síti TCP/IP [20]

Obecně struktury TCP a IP paketů jsou stejné – hlavička a data, ale struktury jejich hlaviček jsou různé. TCP pakety obsahují informace pomocí kterých lze určit pořadí balíčků, jejich celistvost a jejich cílovou aplikaci (pomocí portu) [19].

2.2.4 ISO-on-TCP

ISO-on-TCP je nadstavbou pro TCP, umožňující realizaci principů přenosu dat podle ISO Transport protokolu. Pomocí daného protokolu lze zajistit přenos dat ve formě paketů stejné délky, nazývaných *Transport Protocol Data Unit (TPDU)* [21].



Obr. 5. Pořadí protokolů, používaných při komunikaci s PLC společnosti Siemens

2.2.5 S7 Protocol

S7 Protocol je speciální protokol, vyvíjený společností Siemens a používaný při komunikaci s PLC řady S7-300/400. Používá se k výměně dat mezi PLC a přistoupení k paměti PLC jinými zařízení. Pakety daného protokolu se nazývají *Protocol Data Unit* (PDU) a každý z nich představuje příkaz nebo reakci na něj.

Struktura PDU má 4 elementy:

- Hlavičku;
- Sadu parametrů;
- Data parametrů;
- Blok s ostatními daty

Poslední dva elementy nejsou povinné a mohou být zanechané [13] [22].

3 PROGRAMOVACÍ JAZYK C# A SOFTWAREVÁ PLATFORMA .NET FRAMEWORK

3.1 Common Language Infrastructure

Common Language Infrastructure (CLI) je standard, popisující prostředí, které umožňuje použití různých programovacích jazyků při návrhu aplikace a zajišťuje zprovoznění a běh takové aplikace na různých platformách bez potřeby přepisovat její zdrojový kód. Základními aspekty CLI jsou:

- **Common Type System** (CTS) – sada pravidel pro programovací jazyky pomocí které lze zajistit plynulou interakci datových typů nebo objektů, napsaných v různých jazycích mezi sebou.
- **Metadata** – referenční informace o struktuře programu a jeho metodách. Zajišťují správné a snadné použití těchto metod v jakémkoliv z programovacích jazyků, podporovaných prostředím.
- **Common Language Specification** (CLS) – sada standardů, kterým musí vyhovovat programovací jazyk, aby byla zajištěná plynulá spolupráce s ostatními CLS-jazyky. Je podmnožinou CTS.
- **Virtual Execution System** (VES) - zajišťuje překlad programu do strojového kódu té platformy, na které ten program byl spuštěn. Překládá program na základě poskytnutých metadat a specifikací platformy [23].

3.2 Platforma .NET Framework

.NET Framework je implementaci CLI na operačních systémech Microsoft Windows. Je složen z dvou velkých částí: *Common Language Runtime* (CLR) a *Framework Class Library* (FCL).

3.2.1 Common Language Runtime

CLR je virtuální prostředí, poskytující různé nástroje pro práci s kódem programů. Při kompilaci programu jeho zdrojový kód se nejdříve překládá do *Common Intermediate Language* (CIL) – společného mezijazyka, do kterého se překládají programy, napsané v programovacích jazycích podporovaných prostředím .NET Framework. Během kompilace se vytváří i metadata programu.

Jakmile program bude spuštěn poprvé, proběhne jeho překlad z CIL do strojového kódu na základě vlastností dané platformy – operačního systému a procesoru. Překlad kódu je zajištěn *just-in-time* (JIT) překladačem. Důvodem k použití takového systému je snaha získávat co nejprizpůsobenější pro jakoukoliv platformu program bez potřeby jeho manuální úpravy.

CLR zajišťuje řízený běh kódu programu, což znamená že jednotlivé příkazy programu se vykonávají prostřednictvím CLR, ne přímo na procesoru počítače. Takovým způsobem jsou zajištěné bezpečnost a spolehlivost běhu programů, včetně včasného zpracování výjimek, např. dělení nulou.

Další vlastnosti CLR je možnost plynulé spolupráci mezi částí programu, napsanými v různých jazycích. Například, nějaká z metod programu, napsaného v jazyce VB.NET, může používat moduly z knihovny napsané v jazyce C#.

Garbage Collector (GC) je další důležitou službou v CLR. Při vytvoření nových objektu se alokuje místo v operační paměti počítače, kam pak té objekty budou uloženy. Pokud se s objektem již nebude pracovat, je nutné uvolnit to místo v paměti, aby operační systém mohl jeho využít k jiným účelům, pokud bude potřeba. Jestli místo nebude uvolněno a budou ztraceny všichni ukazatele, pracující s tím místem v paměti, objeví se únik paměti (anglicky *memory leak*) a program již nebude schopen tu část paměti použít ani uvolnit. V jazyce C nebo C++ zajištění správného alokování a uvolnění paměti je povinností programátora. CLR pomocí GC dělá tuto práci automaticky [24].

3.2.2 Framework Class Library

FCL – sada knihoven, obsahujících třídy, rozhraní, delegáty a typy hodnot určených k usnadnění a optimalizaci procesu vyvíjení aplikací. FCL obsahuje třídy, realizující základní funkce:

- realizace základních typů hodnot a výjimek;
- metody zapouzdření;
- práce s V/V;
- přístup k informacím o již použitých v aplikaci třídách;
- vyvolávání bezpečnostní kontroly;
- zajištění toku dat mezi serverem a klientem [24].

3.2.3 Windows Presentation Foundation

Windows Presentation Foundation (WPF) – knihovna tříd prostředí .NET Framework, určených pro vytvoření uživatelských rozhraní k aplikacím. Klíčovou vlastností WPF je to, že její grafické elementy jsou založené na vektorové grafice, což znamená že výsledné rozhraní nebude závislé na rozšíření obrazovky počítače. Pro snadné vyvíjení rozhraní a jeho spojení s logikou programu se používá *Extensible Application Markup Language* (XAML) – značkovací jazyk, vyvíjený společností Microsoft na základě jazyka XML [24].

3.3 Krátký popis vlastností programovacího jazyka C#

C# je vysokoúrovňový objektově orientovaný a typově bezpečný programovací jazyk, pomocí kterého lze vytvářet různé typy programů: od klientských aplikací pro operační systém Windows do aplikací, pracujících s databázemi. Jakýkoli program, napsaný v tomto jazyce, vyžaduje přítomnost .NET Frameworku nebo jeho analogu pro jeho spuštění. Jazyk C# se vyvíjí v společnosti Microsoft, spolu s .NET Framework. Syntaxe jazyka C# je dost podobná jazykům C, C++ a Java.

Při vyvíjení jazyka byl kladen důraz na zajištění robustnosti a trvanlivosti každého programu, vytvořeného při jeho použití. C# využívá funkcí, nabízených prostředím .NET Framework, ke kontrole využití paměti programem (GC) a detekci a zpracování chyb při jeho běhu. Navíc k tomu, C# je typově bezpečný jazyk se statickým typováním, což znamená že v něm není možné pracovat s neinicializovanými proměnnými nebo provádět nekontrolované přetypování proměnných. Nicméně, v C#, při použití modifikátoru *unsafe*, lze pracovat přímo s pamětí počítače, stejně jako v C++ [25].

II. PRAKTICKÁ ČÁST

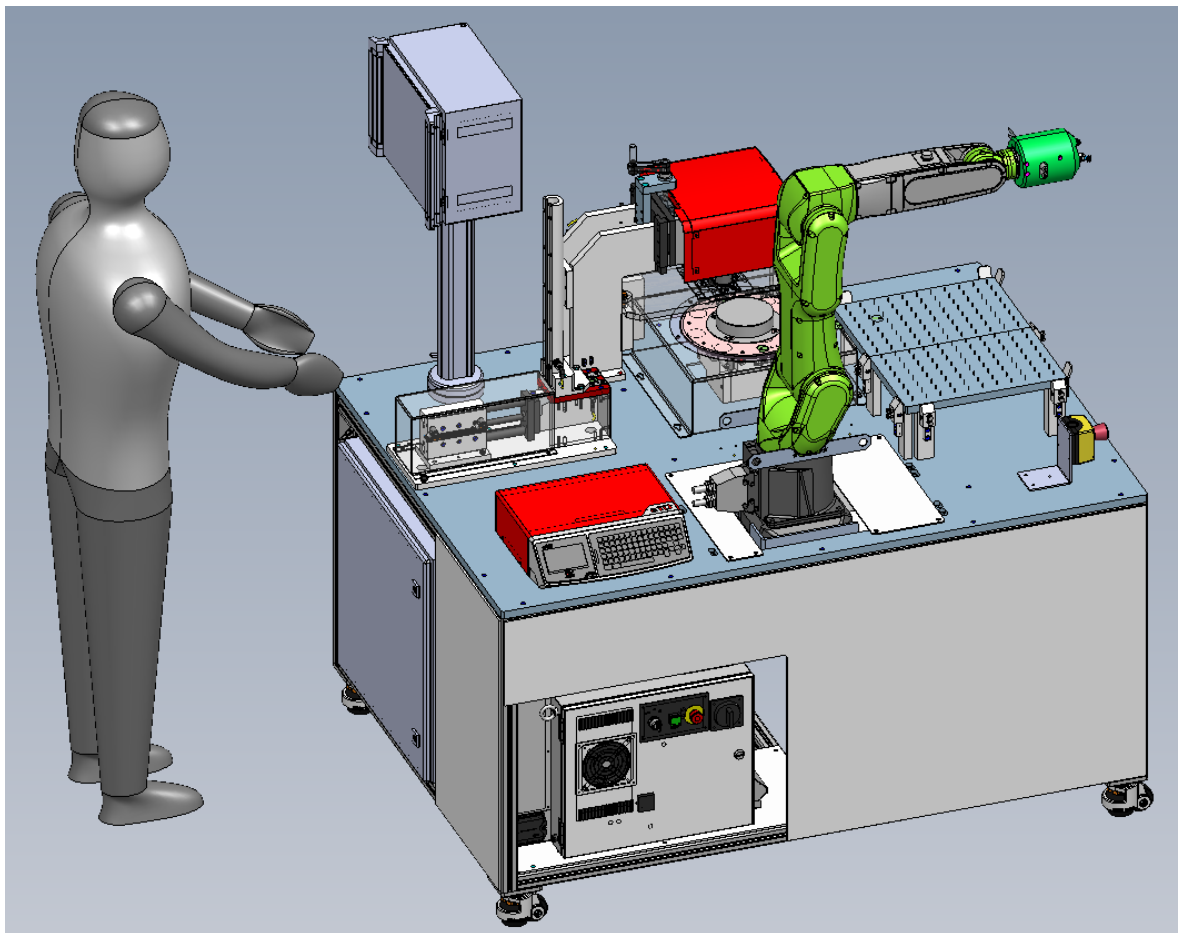
4 POPIS GRAVÍROVACÍHO STROJE

Gravírovací stroj představuje několik zařízení, která mezi sebou spolupracují. Jeho součástmi jsou:

- PLC s HMI rozhráním;
- systém značení;
- kolaborativní robot;
- osobní počítač;
- podavač štítků;
- rotační stůl

a další prvky. Návrh stroje, výběr součástí a jejich případné programování byli provedeny společností AWL-Techniek CZ s.r.o a nejsou součástí dané práce.

Řídící jednotkou celkového stroje je PLC. Pomocí Ethernetového kabelu je PLC propojeno s osobním počítačem, na kterém běží program vytvořený v rámci této bakalářské práce.



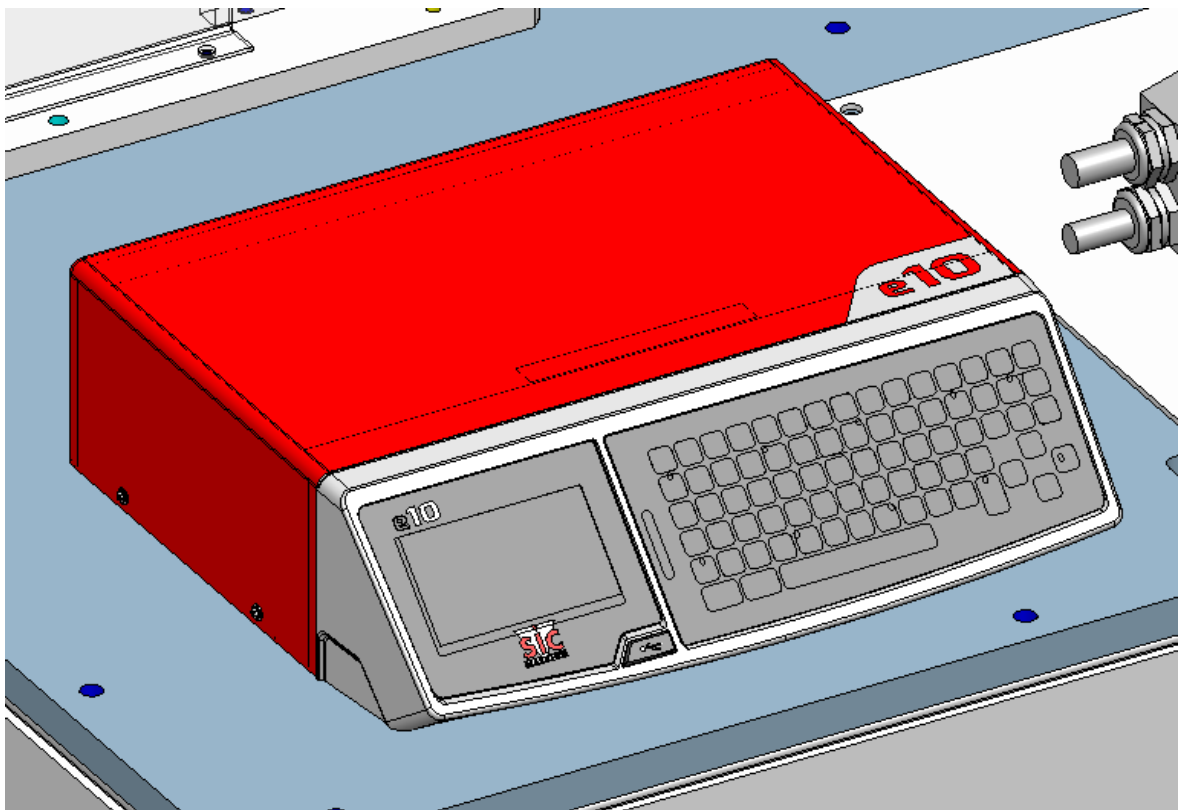
Obr. 6. Celkový pohled na stroj

4.1 Popis používaného PLC

Hlavní součástí daného stroje je PLC *CPU 1516F-3 PN/DP* z řady *S7-1500F* společnosti Siemens. V roli uživatelského rozhraní se používá ovládací panel *SIMATIC HMI TP900 Comfort* jehož funkcí je ovládaní celkového stroje a případné poskytování informací o jeho stavu. PLC obsahuje další I/O moduly určené ke komunikaci s dalšími částmi stroje prostřednictvím *PROFINET* sběrnice a modul, určený k řízení pneumatických ventilů.

4.2 Popis systému značení firmy SIC MARKING

Systém značení se skládá y řídicí jednotky *e10* a mechanické hlavy s hrotem *c153*. Operační systém kontroléru poskytuje širokou škálu možností kontroly procesu značení. Například, při vygravírování textových řádků lze zvolit písmo, velikost a úhel otočení textu. Systém je také schopen vygravírovat QR nebo čárové kódy a jednoduché obrázky (logo firmy). Kontrolér má vlastní uživatelské rozhraní s membránovou klávesnicí určenou pro manuální ovládaní systému značení. Daný systém lze ovládat i vzdáleně, a to posíláním příkazů přes sběrnici, což umožňuje začlenit daný systém do automatizované linky.



Obr. 7. Řídicí jednotka e10

4.3 Popis používaného robota

V roli pick-and-place manipulátoru se používá kolaborativní robot *CR-7iA/L* firmy FANUC. Daný model robota má 6 rotačních kloubů a 6 stupňů volnosti. Manipulátor je schopen přenášet předměty o hmotnosti do 7 kg. Pokud senzory robota detekují vnější sílu, působící na manipulátor, robot se zastaví, aby neublížil operátorovi.

Daný robot je řízen kontrolérem *R-30iB Mate*, který řídí pohyby robota a provádí diagnostiku jeho okamžitého stavu na základě informací, získaných ze senzorů robota. Ke kontroléru je také připojen Teach Pendant.

Efektor představují dvě vakuové přísavky, pomocí kterých se uskutečňuje manipulace se štítky. Kontrolér robota je řízený pomocí PLC, které určuje, jaký z programů robota musí být spuštěn v určitý okamžik.

4.4 Další částí stroje

4.4.1 PC

Pro práci se strojem lze použít jakýkoli osobní počítač s operačním systémem *Windows 7* nebo *Windows 10*. Použití počítače s jiným operačním systémem je teoreticky možné, pokud na daném systému lze zprovoznit prostředí .NET Framework 4.5. V takovém případě existuje ovšem možnost výskytu chyb v běhu programu, protože program byl otestován jenom na systémech uvedených na začátku tohoto odstavce.

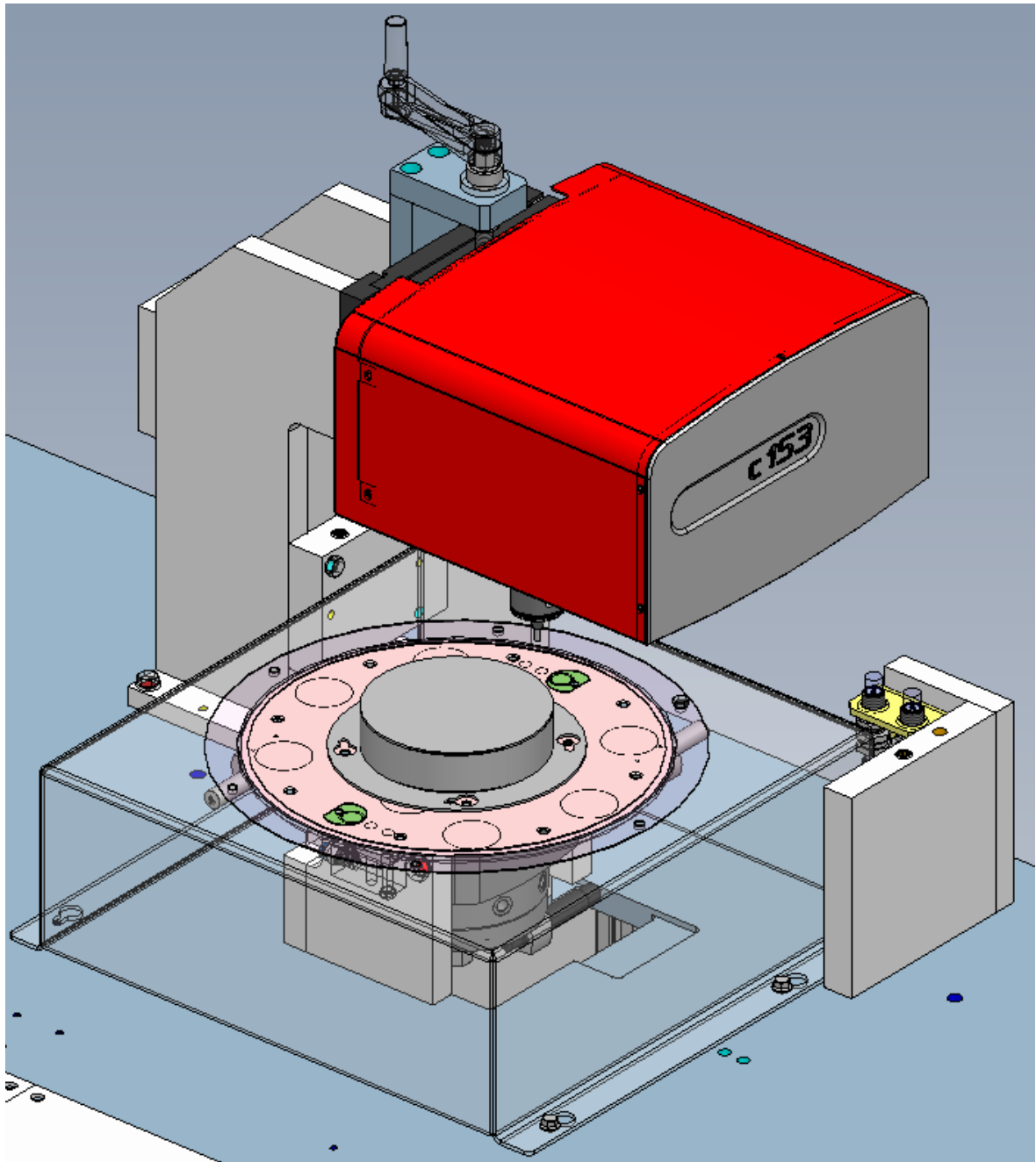
Propojení s PLC je uskutečněno prostřednictvím Ethernetového kabelu *RJ-45* s konektorem *8P8C*.

4.4.2 Podavač štítků

Podavač štítků je mechanismus zajišťující dodání prázdných štítků pro jejich další vyzvednutí robotem. Je reprezentován zásobníkem ve formě sloupu, umístěným nad posuvným kovovým blokem. Daný blok se pohybuje mezi dvěma pozicemi při podání štítku. Na místě odběru štítků je umístěn bezkontaktní indukční snímač, který detekuje jejich přítomnost. Jakmile je zjištěno, že štítek není přítomen, tak se spustí cyklus podání nového prázdného štítku.

4.4.3 Rotační stůl

Rotační stůl se používá ke výměně štítků mezi robotem a mechanickou částí gravírovacího systému. Na stole jsou 2 pozice pro vkládání štítků, přičemž každá pozice je vybavená vakuovou přísavkou pro fixaci polohy štítku během otáčení stolu a procesu gravírování.



Obr. 8. Rotační stůl a mechanická hlava s hrotem c153

4.4.4 Plato na štítky

Místo, na které se odkládají hotové štítky. Je to dřevěná základna se zčásti zatlučenými hřebíky v počtu 176 kusů – 11 sloupců a 16 řádků. Každý hřebík představuje pozici, na kterou robot vkládá hotový štítek. Pozice se vyhodnocuje pomocí PLC.

4.4.5 Vzhled štítku

Hliníkové štítky, se kterými pracuje stroj, mají kulatý tvar o poloměru 15 mm a tloušťce 2 mm. Vedle odřezu je umístěn otvor o průměru 0,5 mm který se používá při fixaci štítku na kabelu.



Obr. 9. Hliníkový štítek

5 ANALÝZA POŽADAVKŮ SPOLEČNOSTI AWL-TECHNIEK CZ S.R.O.

Daná společnost se zabývá návrhem a vytvořením automatizovaných svařovacích linek, jejichž jednotlivé části jsou propojené mezi sebou pomocí kabelů na základě schémat zapojení. Každý kabel má svoje vlastní označení na tomto schématu, aby bylo možné rozpoznat kam tento kabel patří a jak má být zapojený. Na označení každého kabelu je potřeba použít 2 až 3 štítky v závislosti na jeho délce. Průměrné množství kabelů v lince dosahuje 300 kusů což znamená že je potřeba připravit 600 až 900 štítků.



Obr. 10. Označení kabelů pomocí hliníkových štítků

Předtím než byl postaven stroj uvedený v kapitole 4, příprava štítků byla zcela manuální práci, za výjimkou procesu gravírování. Operátor musel zafixovat sadu s prázdnými štítky pod mechanickou hlavou gravírovacího systému, pak zadat texty pro každý štítek a potom odebrat hotové štítky po ukončení značení. Vzhledem k potřebnému množství štítků je to časově náročná práce, vyžadující vysokou míru pozornosti. Nově postavený stroj řeší problémy ohledně manipulaci se štítky, ale aby stroj měl informace o tom, co je potřeba vy-

gravírovat na štítcích, je nutné navrhnout program, který bude schopen získat potřebnou informaci a zapsat ji do paměti stroje.

5.1 Požadavky firmy

Dále jsou uvedeny základní požadavky firmy a jejich příslušná řešení:

- **Rychlá extrakce potřebných dat ze speciálních souborů s příponou ‘.txt’ a kontrola jejich správnosti.** Dané soubory obsahují jednotlivé textové řetězce představující hesla, které pak budou vygravírovány na štítcích. Program kontroluje náplně, délky a počet jednotlivých řetězců v každém řádku souboru. Tím je zajištěna správnost vstupních dat. Rychlost jejich zpracování je závislá na výkonu systému, na kterém ten program běží. Běžný počítač to zvládá maximálně za několik milisekund.
- **Zajištění celistvosti přijatých dat v paměti PLC.** Program porovnává symboly v paměti PLC se symboly ve své paměti na totožných pozicích. V případě neshody program znovu pošle správný symbol do paměti PLC a zase ho zkontroluje.
- **Možnost sledování stavu gravírování.** V programu jsou realizovány prostředky ke sledování stavu každého štítku. Na základě informací o všech štítcích program je schopen rozhodnout o stavu gravírování každé série štítků.
- **Jednoduchost použití programu.** Základní uživatelské rozhraní programu obsahuje jenom nejdůležitější funkce a informace. V případě výskytu chyby program buď pokusí ji vyřešit samostatně nebo popíše co ji způsobilo. K dispozici jsou anglická a česká verze uživatelského rozhraní.

5.2 Požadavky operátorů

Dané požadavky byly zformulovány na základě zpětné vazby operátorů stroje, které ten program používají:

- **Možnost tištění několika menších sad štítků na jedno plato.** Algoritmus nahrávání dat do PLC byl rozšířen o možnost přidávání dalších dat z jiných souborů, k již nahaným. Program ukazuje, na které pozice lze přidat extrahovaná data na základě jejich množství a zbývajících kapacit těchto pozic.
- **Možnost změnit počet kopií štítků.** Přidaná možnost zkontrolovat, jak program zpracoval data ze vstupního souboru. Po stisku tlačítka „Ukázat Extrahovaná Data“

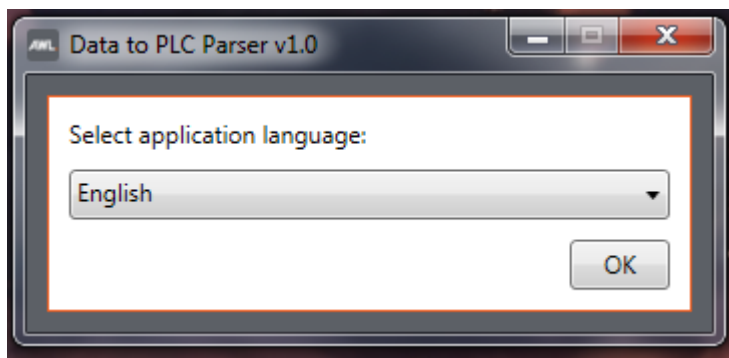
se objeví okno s informací o štítcích, ve kterém je lze přeurčit, přidat další nebo smazat stávající před nahráváním do paměti PLC.

- **Lepší zvýraznění informací o stavu štítků.** Bylo přidáno barevné označení stavů štítků, což umožnilo zjišťování potřebné informace na větší vzdálenosti od stroje.
- Další drobné změny, tykající se jednotlivých částí uživatelského rozhraní.

6 UŽIVATELSKÉ ROZHRAŇÍ PROGRAMU PRO PRACÍ S GRAVÍROVACÍM STROJEM

6.1 Volba jazyka rozhraní

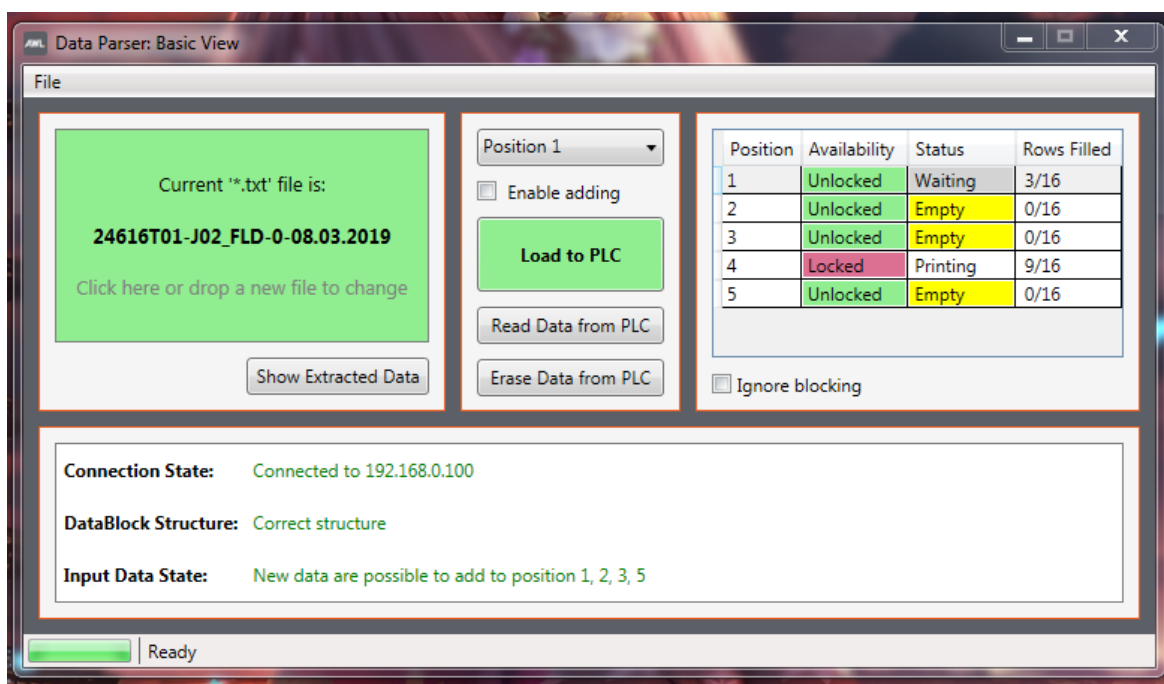
Po spuštění programu se objeví okno s volbou jazyka rozhraní. Nabízené jazyky jsou čeština a angličtina.



Obr. 11. Volba jazyka rozhraní

6.2 Hlavní okno

Hlavní okno aplikace se skládá se 6 částí: *Lišta s menu*, *Panel vstupního souboru*, *Hlavní ovládací panel*, *Tabulka stavů pozic*, *Tabulka stavu aplikace* a *Stavový řádek*.

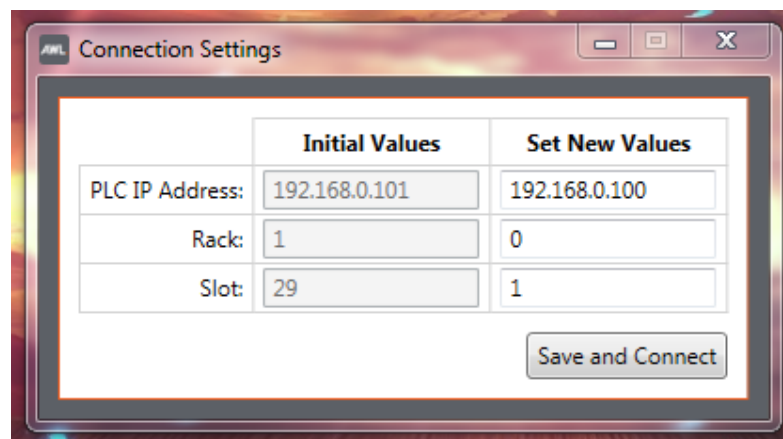


Obr. 12. Hlavní okno programu

- **Lišta s menu** má jednu nabídku – *File*, obsahující dva příkazy: *Open* a *Exit*. Pomocí prvního příkazu lze zvolit soubor s daty, pomocí druhého lze zavřít aplikaci. Dané příkazy podporují volbu pomocí klávesových zkratk.
- **Panel vstupního souboru** má v sobě obdélníkovou oblast (Drop oblast) do které se vkládají soubory s daty. Po stisknutí tlačítka *Show Extracted Data* se objeví okno s daty z nahraného souboru.
- **Hlavní ovládací panel** se používá při provedení operací s PLC. Pomocí tohoto panelu lze nahrávat, číst nebo mazat data v paměti PLC.
- **Tabulka stavů pozic** poskytuje informace o procesů gravírování. Nabídka *Ignore blocking* umožňuje operátorovi měnit data na blokových pozicích.
- **Tabulka stavu aplikace** poskytuje operátorovi informace o aktuálním stavu aplikace, aby bylo možné řešit případné chyby, které by se mohli vyskytovat během chodu aplikace.
- **Stavový řádek** poskytuje informaci o běžících procesech.

6.3 Nastavení připojení

Okno, obsahující 3 položky: *PLC IP Address*, *Rack* a *Slot*. Před zavřením okna nebo po stisknutí tlačítka *Save and Connect* program zkontroluje, jestli nové parametry jsou správné.



Obr. 13. Okno nastavení parametrů připojení

6.4 Nastavení struktury databáze v PLC

Okno s velkým počtem parametrů, pomocí kterých lze přeurčit referenční strukturu datového bloku.

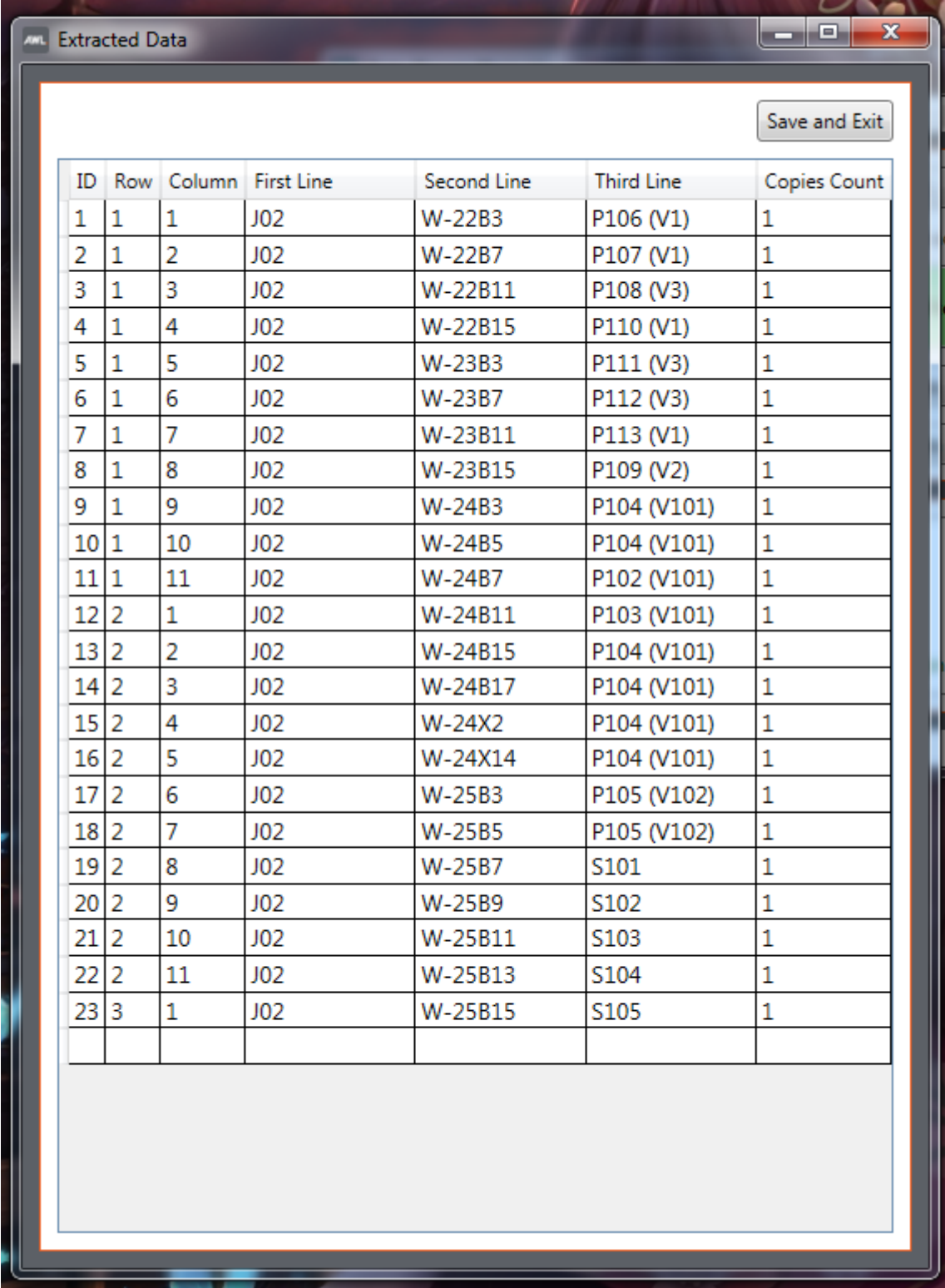
	Initial Values	Set New Values
PLC DB Number:	100	
Plate Size:	Columns: 11 Rows: 16	Columns: Rows:
Max. Copies:	4	
New Plate Offset:	12726	12721
Plate Name Offset:	2	
'Order Placed' Offset:	54	49
'Finished Labels' Offset:	56	51
'Copies Info' Offset:	58	53
First Line Offset:	60	55
Second Line Offset:	82	77
Third Line Offset:	104	99
Next Label Distance:	72	
Max. Plate Name Length:	50	45
Max. Label Line Length:	20	
Allowed Label Line Length:	15	

Save and Evaluate

Obr. 14. Okno s parametry struktury datového bloku

6.5 Náhled dat ze zpracovaného souboru

Jakmile program ukončí extrakci dat ze vstupního souboru, operátor má možnost je zkontrolovat, případně i změnit. So stisknutí tlačítka *Save and Exit* bude provedená kontrola, jestli program detekoval změnu vstupních dat. Pokud ano, bude provedená kontrola jejich správnosti.



The screenshot shows a window titled "Extracted Data" with a "Save and Exit" button in the top right corner. The window contains a table with the following data:

ID	Row	Column	First Line	Second Line	Third Line	Copies Count
1	1	1	J02	W-22B3	P106 (V1)	1
2	1	2	J02	W-22B7	P107 (V1)	1
3	1	3	J02	W-22B11	P108 (V3)	1
4	1	4	J02	W-22B15	P110 (V1)	1
5	1	5	J02	W-23B3	P111 (V3)	1
6	1	6	J02	W-23B7	P112 (V3)	1
7	1	7	J02	W-23B11	P113 (V1)	1
8	1	8	J02	W-23B15	P109 (V2)	1
9	1	9	J02	W-24B3	P104 (V101)	1
10	1	10	J02	W-24B5	P104 (V101)	1
11	1	11	J02	W-24B7	P102 (V101)	1
12	2	1	J02	W-24B11	P103 (V101)	1
13	2	2	J02	W-24B15	P104 (V101)	1
14	2	3	J02	W-24B17	P104 (V101)	1
15	2	4	J02	W-24X2	P104 (V101)	1
16	2	5	J02	W-24X14	P104 (V101)	1
17	2	6	J02	W-25B3	P105 (V102)	1
18	2	7	J02	W-25B5	P105 (V102)	1
19	2	8	J02	W-25B7	S101	1
20	2	9	J02	W-25B9	S102	1
21	2	10	J02	W-25B11	S103	1
22	2	11	J02	W-25B13	S104	1
23	3	1	J02	W-25B15	S105	1

Obr. 15. Náhled na okno, obsahující data ve vyrovnávací paměti programu

6.6 Náhled dat v paměti PLC

V daném okně jsou ukázána data jedné z pozic v paměti PLC. Dané okno je používáno pro podrobné sledování procesu gravírování štítků. Hotové kopie štítků jsou označené zeleně pro lepší a rychlejší orientaci operátora, pracujícího s programem.

AWL Plate name: 24616T01-J02_FLD-0-08.03.2019

Finished Labels Count: 2
Total Label Count: 69

ID	Row	Column	First Line	Second Line	Third Line	Order Placed	Finished Labels	Copies Count
1	1	1	J02	W-22B3	P106 (V1)	1	1	1
2	1	2	J02	W-22B7	P107 (V1)	1	1	1
3	1	3	J02	W-22B11	P108 (V3)	0	0	1
4	1	4	J02	W-22B15	P110 (V1)	0	0	1
5	1	5	J02	W-23B3	P111 (V3)	0	0	1
6	1	6	J02	W-23B7	P112 (V3)	0	0	1
7	1	7	J02	W-23B11	P113 (V1)	0	0	1
8	1	8	J02	W-23B15	P109 (V2)	0	0	1
9	1	9	J02	W-24B3	P104 (V101)	0	0	1
10	1	10	J02	W-24B5	P104 (V101)	0	0	1
11	1	11	J02	W-24B7	P102 (V101)	0	0	1
12	2	1	J02	W-24B11	P103 (V101)	0	0	1
13	2	2	J02	W-24B15	P104 (V101)	0	0	1
14	2	3	J02	W-24B17	P104 (V101)	0	0	1
15	2	4	J02	W-24X2	P104 (V101)	0	0	1
16	2	5	J02	W-24X14	P104 (V101)	0	0	1
17	2	6	J02	W-25B3	P105 (V102)	0	0	1
18	2	7	J02	W-25B5	P105 (V102)	0	0	1
19	2	8	J02	W-25B7	S101	0	0	1
20	2	9	J02	W-25B9	S102	0	0	1
21	2	10	J02	W-25B11	S103	0	0	1
22	2	11	J02	W-25B13	S104	0	0	1
23	3	1	J02	W-25B15	S105	0	0	1
24	3	2				0	0	0
25	3	3				0	0	0
26	3	4				0	0	0
27	3	5				0	0	0
28	3	6				0	0	0
29	3	7				0	0	0

Obr. 16. Okno s daty v paměti PLC

7 VLASTNOSTI PROGRAMU PRO PRÁCI S GRAVÍROVACÍM STROJEM

7.1 Popis nástroje Snap7

Snap7 je multiplatformní sada nástrojů, určených k výměně informací se zařízeními firmy Siemens prostřednictvím technologií *Ethernet*. Zajišťuje skoro úplnou podporu *S7 Protocolu* na jehož základě probíhá komunikace mezi PLC a počítačem.

Automatizační zařízení firmy Siemens používají vlastní model síťové architektury při komunikaci mezi sebou. Každý prvek sítě pracuje na základě jedné ze třech rolí – *Client*, *Server* a *Partner*:

- *Client* může jenom posílat dotazy;
- *Server* může jenom odpovídat na dotazy;
- *Partner* je kombinací obou. Je podoben *peeru* v počítačových sítích.

Je zřejmé, že pro smysluplnou komunikaci je potřeba aby každý uzel sítě měl přidělenou určitou roli a na její základě přesně vykonával svoji činnost. *Snap7* nabízí realizaci každého typu uzlu daného modelu.

7.1.1 Snap7 Client

Program v roli klientu může navázat připojení a posílat dotazy do PLC, které převezme roli serveru. Pomocí *Snap7* lze pracovat s prvky v paměti PLC, kontrolovat jeho běh, spravovat zabezpečení programu heslem atd. Veškerá komunikace běží na pozadí, aby byl zajištěn nepřetržitý běh programu na straně klientu. Je důležité poznamenat že novější modely PLC (např. řády *S7 1200/1500*) používají modifikovaný *S7 Protocol*. Při práci s takovými PLC jsou podporovaný jenom základní operace čtení a zápisu do jejich paměti.

7.1.2 Snap7 Server

Pomocí modulu *Snap7 Server* program může vykonávat roli serveru a je schopen komunikovat s několika klienty zároveň. Ve své podstatě je to částečná simulace PLC. Na daném serveru jsou realizované jenom některé funkce jako například nastavení data a času, čtení a zápis do paměti „PLC“ nebo simulace kontroly běhu „PLC“. Na dotaz obsahující nepodporovanou funkci server neodpoví nic. Pro komunikaci s každým klientem se vytváří vlastní

vlákno, které je k dispozici do momentu odpojení klienta. Existuje možnost zablokovat blok paměti serveru, ve kterém bude provedená manipulace s daty do její ukončení. Tím se dá zajistit postupné zpracování dat.

7.1.3 Snap7 Partner

Daný modul lze použít při komunikaci typu peer-to-peer mezi dvěma uzly. Jeden z partnerů, který požádá o připojení, bude označen jako *active*, což znamená že on převzme roli klientu. Ten druhý, co požadavek přijme, bude označen jako *passive* a bude fungovat jako server. Mezi partnery jsou podporovány synchronní a asynchronní modely přenosu dat.

Tento typ propojení je vhodné použít, pokud je potřeba zajistit sbírání velkého množství dat z několika PLC, zařazených do jedné výrobní linky, přičemž každé PLC vytváří data na základě dat od předchozích PLC.

7.2 Popis základních funkcí programu „Data to PLC Parser“

Program představuje sadu různých funkcí a algoritmů, jejichž základním účelem je zajistit plynulou spolupráci operátora a gravírovacího stroje. Jednou z nejdůležitějších částí programu je pole, obsahující příznaky základních funkcí, určených k výměně dat mezi PC a PLC. Na základě hodnot těchto příznaků program vykonává potřebné činnosti. Většina funkcí se vykonává asynchronně, na pozadí, aby byla zajištěná možnost použití programu operátorem v libovolný moment času.

Jednotlivé typy činností programu jsou:

- **Připojení k PLC.** Funkce, pomocí které probíhá navázání připojení k PLC. Je spuštěná hned po startu programu nebo když program zjistí, že propojení s PLC se přerušilo. Program zkusí se připojit k PLC čtyřikrát, pokud se mu to nepodaří - vyhlásí chybu (viz 7.3.1).
- **Kontrola struktury databáze v PLC.** Jakmile bude navázáno propojení s PLC, program zkontroluje strukturu jeho databáze, a to formou čtení bajtů na určitých pozicích datového bloků v paměti PLC. Program má vlastní referenční hodnoty a porovnává je s těmi, co byli načteny. Pokud alespoň jedná hodnota bude odlišná, program řekne, že struktura databáze v paměti PLC je chybná (viz 7.3.2). Po

úspěšné kontrole struktury databázi PLC, program spočítá, kolik je v té databázi pozic – sad štítků.

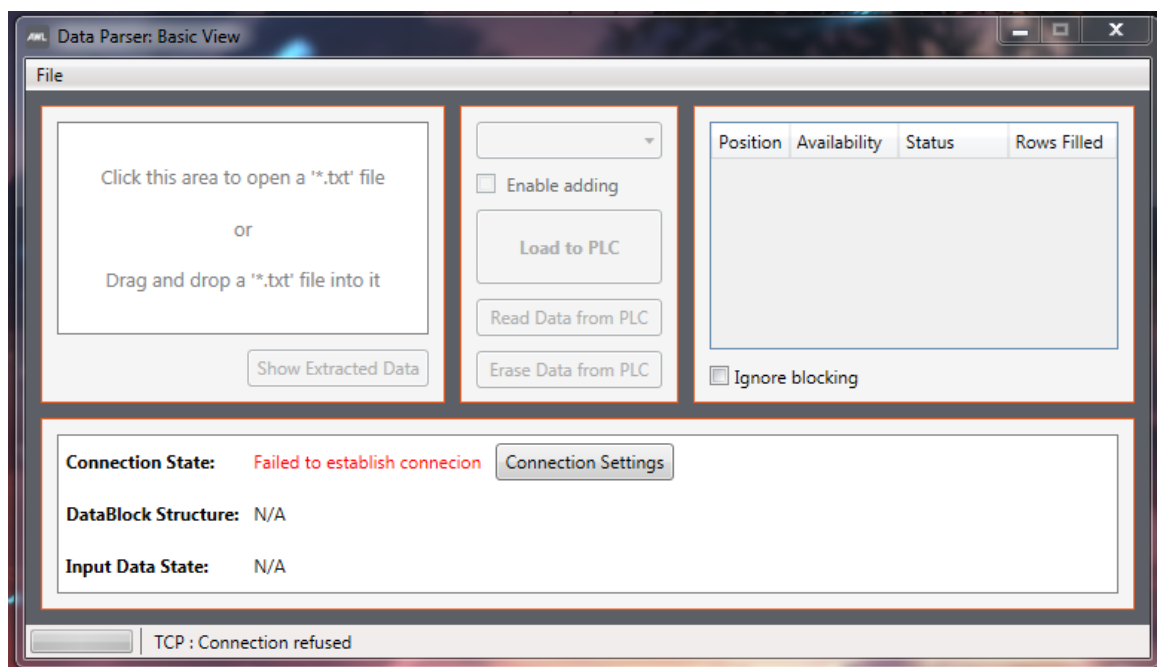
- **Vytvoření databáze na PC.** Pokud struktura databáze v paměti PLC je správná, program vytvoří svoji vlastní databázi, se stejnou strukturou a stejným počtem pozic. Do té databázi se budou průběžně ukládat data z paměti PLC, ale jenom na požadavek operátora z důvodu minimalizací množství posílaných dat. Po úspěšném vytvoření databázi program už je připravený k extrakci a výměně dat s PLC.
- **Zpracování vstupních souborů.** Až operátorem bude zvolen soubor ke zpracování, program, než začne extrakci dat, nejdříve ověří, jestli ten soubor je správný: bude zkontrolována jeho přípona, velikost v bajtech a vnitřní struktura. U souborů se správnou strukturou každý řádek obsahuje informace, které jsou uspořádané od třech do čtyř sloupců pomocí tabulátorů. V případě úspěchu, program je načte a uloží do vyrovnávací paměti, při neúspěchu – vyhlásí chybu (viz 7.3.3).
- **Zápis získaných dat do paměti PLC.** Přenos dat z vyrovnávací paměti aplikace do paměti PLC bude zahájen, jakmile operátor si vybere pozici v paměti PLC a stiskne příslušné tlačítko. Data jsou přenášeny po bytech. Program nejdříve posílá jeden byte informaci na určitou pozici v paměti PLC a pak načítá s té pozici ten byte a porovnává je. V případě shody algoritmus pokračuje dál, v jiném případě algoritmus se opakuje, pokud získané byty nebudou stejné. Maximální počet pokusů je zase 4, po jejich vyčerpání program vyhlásí chybu (viz 7.3.4). Během přenosu, data z vyrovnávací paměti aplikace se ukládají i do databázi aplikace. Je důležité vědět, že program dovolí nahrávání dat na zvolenou pozici jen jestli ona nebude blokována nebo jestli operátor manuálně nastaví ignorování programem stavů pozic.
- **Čtení dat z PLC.** Daný algoritmus se spustí na základě požadavku operátora. Program načítá data z paměti PLC a ukládá je do své databázi. Po ukončení procesu se objeví okno s načtenými daty, pomocí kterého lze sledovat stav tištění štítků (viz 6.4).
- **Vymazání dat z PLC.** Další proces, který může být zahájen jen operátorem. Program vymaže data z určité pozici ve své databázi a do paměti PLC na stejnou pozici zapíše nuly. Stejně jako u zápisu, tento proces lze spustit, pokud zvolená pozice není blokována.
- **Sledování procesu gravírování.** Každý unikátní štítek v databázi PLC má 3 čísla, popisující jeho stav – počet kopií, počet štítků v procesu gravírování a počet hoto-

vých štítků. Program načítá této hodnoty z paměti PLC a na jejich základě rozhoduje o stavu gravírování jak jednotlivých štítků, tak i o celé sadě. Čtení těchto hodnot probíhá každou sekundu. Z důvodu optimalizace procesu, program načítá hodnoty jenom z těch pozic v databázi PLC, na kterých jsou přítomná data v databázi aplikace.

7.3 Detekce chyb za běhu programu

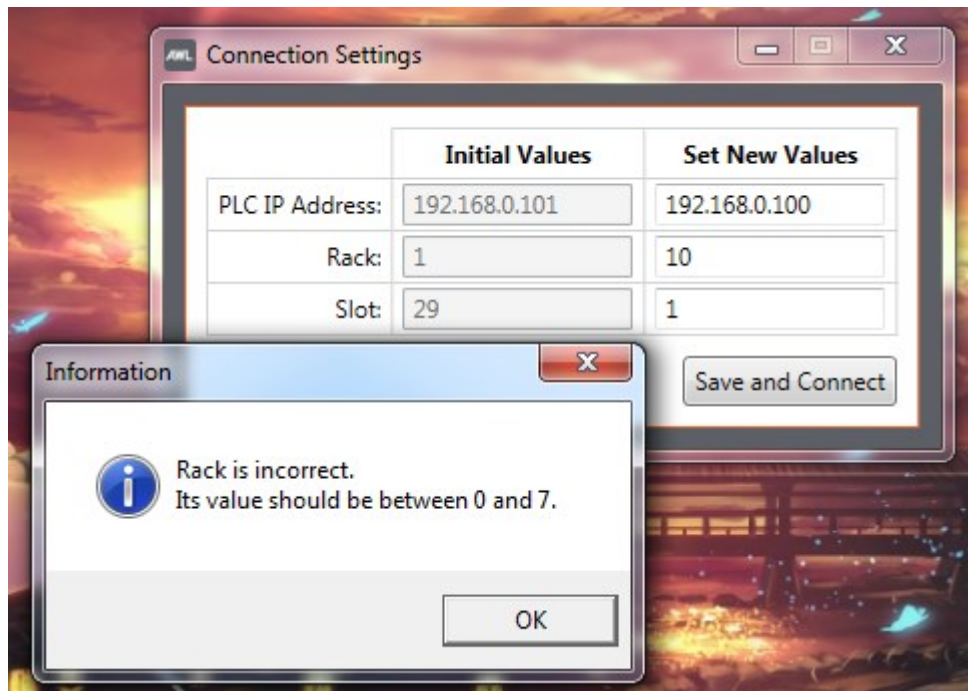
7.3.1 Neúspěšné připojení k PLC

Pokud program nedokáže navázat připojení na PLC, v poli *Connection State* se objeví chybová hláška, spolu s tlačítkem *Connection Settings*.



Obr. 17. Chyba připojení k PLC

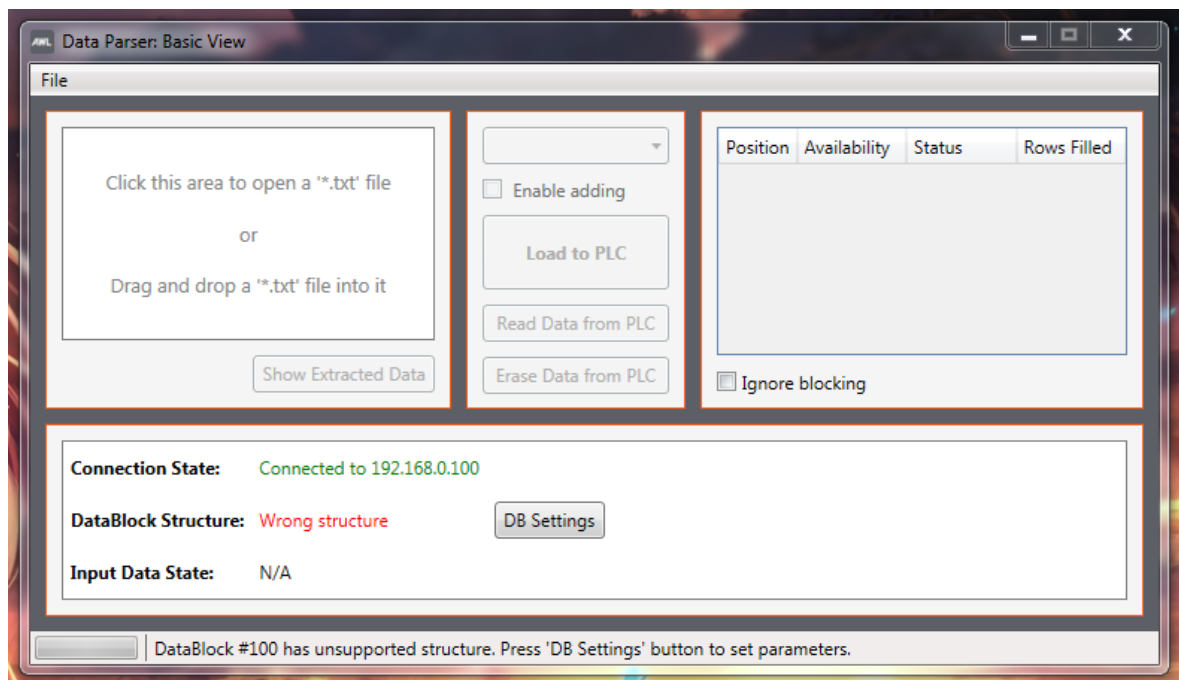
Po jeho stisknutí se objeví okno nastavení parametrů připojení. Operátor může buď ověřit správnost parametrů připojení nebo přeručit je. Jakmile okno bude zavřeno, program se pokusí připojit do PLC znovu a pro připojení bude používat nově nastavené parametry.



Obr. 18. Okno s parametry připojení s hlášením chyby

7.3.2 Chybná struktura databáze v PLC

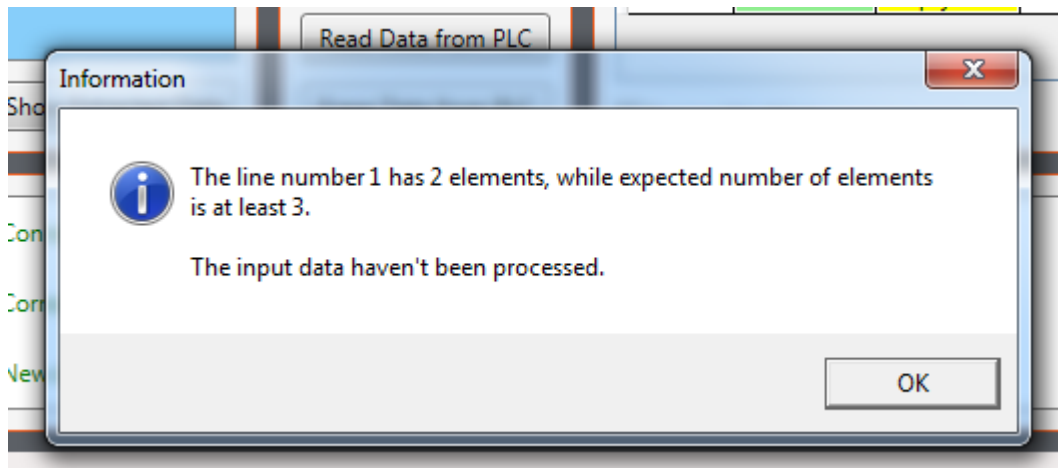
Stejně jak s připojením, pokud selže kontrola struktury datového bloku PLC, – objeví se chybová hláška, spolu s tlačítkem pro přeurčení jeho očekávané struktury.



Obr. 19. Chyba při kontrole struktury datového bloku PLC

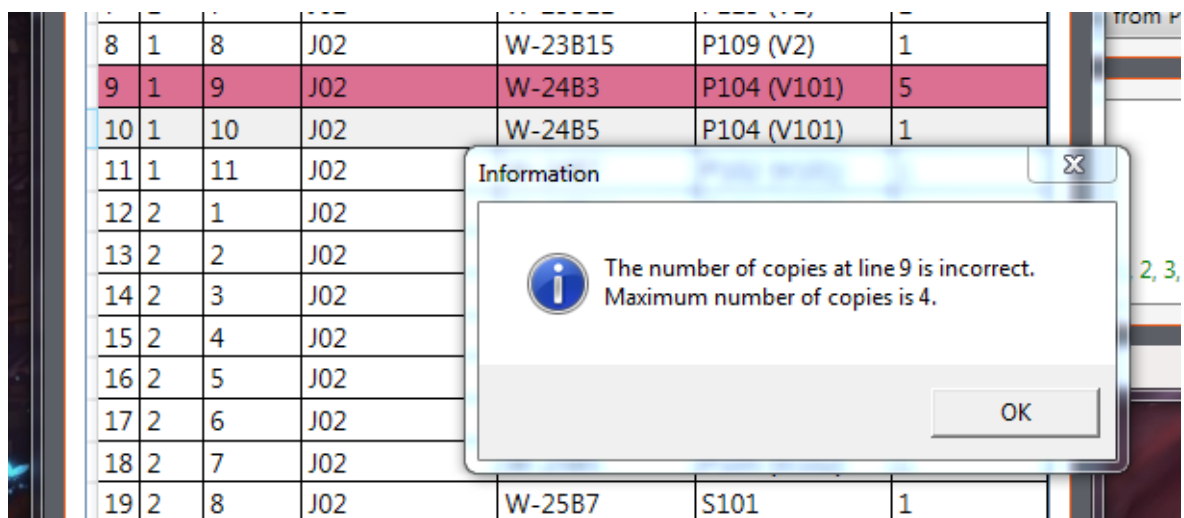
7.3.3 Chyby při zpracování vstupních souborů

Jakmile operátor nahraje špatný soubor, program ho na to upozorní. Nahraný soubor nebude zpracován.



Obr. 20. Příklad chybové hlášky při nahrávání špatného souboru

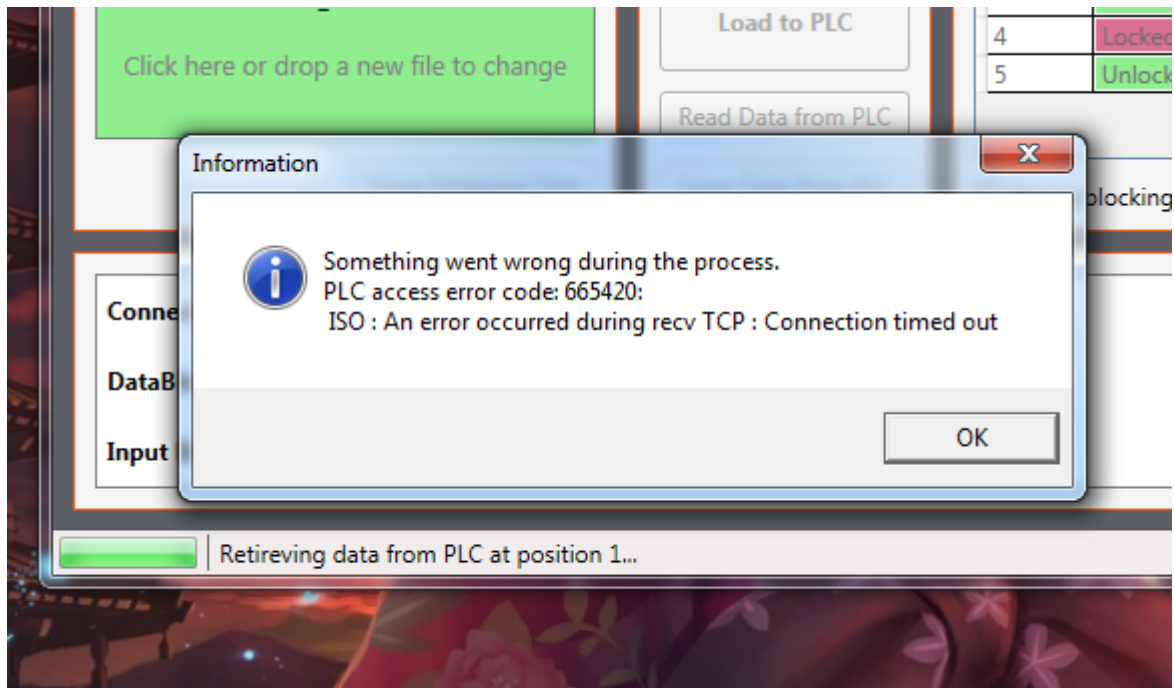
Pokud operátor nahraje správný soubor z hlediska jeho struktury, ale provede chybnou úpravu dat ve vyrovnávací paměti aplikaci, objeví se chybová hláška, popisující to, co způsobilo chybu.



Obr. 21. Chybná úprava vstupních dat operátorem

7.3.4 Chyby přenášení dat

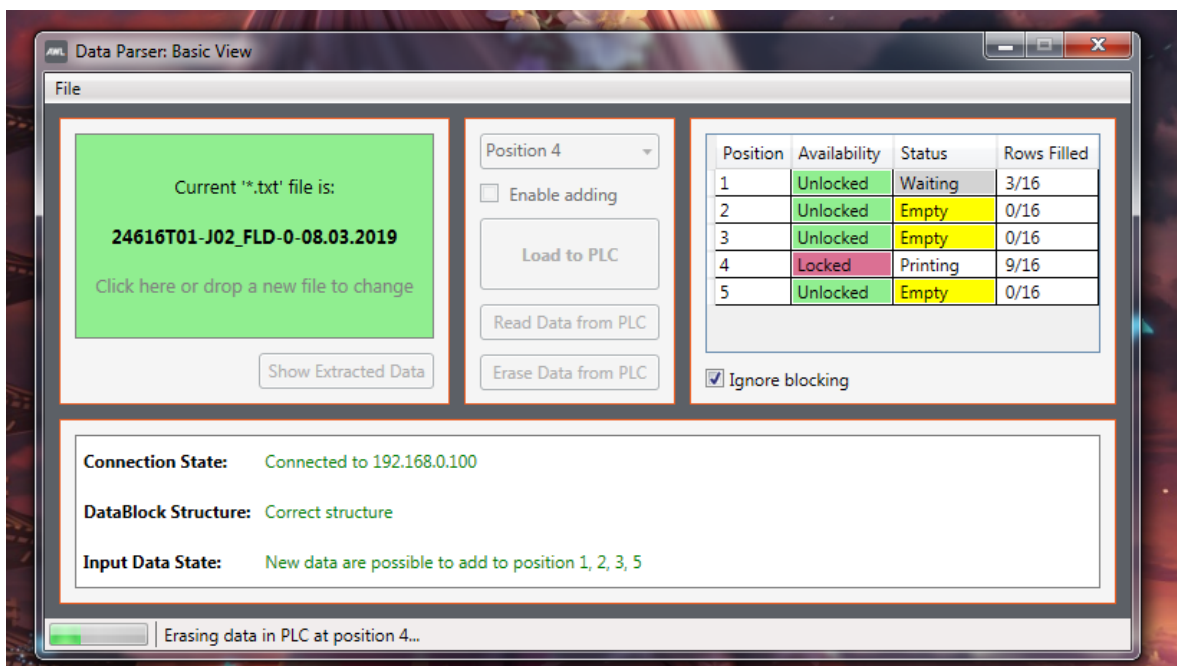
Program je schopen nejenom detekovat chyby během zpracování dat, ale zkoušet je automaticky upravovat tím, že budou vynulovaný příznaky všech funkcí a program se restartuje.



Obr. 22. Detekce neznámé chyby během přenosu dat

7.3.5 Odolnost programu vůči jeho nesprávnému použití

Program automaticky aktivuje nebo deaktivuje určité části svého rozhraní na základě hodnot příznaků základních funkcí. Například nelze zahájit zápis dat do paměti PLC, pokud ten proces už běží.



Obr. 23. Deaktivace částí uživatelského rozhraní

ZÁVĚR

Jednou z příčin vzniku dané bakalářské práce bylo hledání vhodného způsobu zprovoznit již navržený gravírovací stroj, který by mohl vykonávat monotónní práci – gravírování hliníkových štítků. Všichni části stroje už mohli mezi sebou spolupracovat, ale nebylo možné je uvést do nepřetržitého provozu kvůli tomu, že chyběl nástroj, který by mohl zadávat potřebná data pro gravírování. V roli takového nástroje může být i člověk, zadávající data ručně, ale v takovém případě použití stroje by neplnilo základní požadavek při jeho návrhu – uvolnění času pro operátory, aby oni mohli vykonávat jiné druhy práce, vyžadující přítomnost člověka.

Při zpracování teoretické části této práce bylo nalezeno vhodné a rozumné řešení daného problému – volba jazyku C# jako jednoduchého při použití, ale mocného nástroje, pomocí kterého by šlo navrhnout spolehlivou aplikaci, která by měla za účel, jak rychle zpracovávat velké množství dat, tak i posílat je do stroje. Při studiu problematiky propojení osobních počítačů a PLC firmy Siemens bylo řešeno použít již hotovou sadu nástrojů, protože návrh svých vlastních nástrojů by byl z hlediska času příliš náročný a výsledné řešení pravděpodobně mohlo by být nespolehlivé kvůli vysoké míře obtížnosti dané problematiky.

Během zpracování praktické části dané bakalářské práce byla navržena, naprogramována, otestována a uvedena do provozu aplikace, jejíž cílem bylo zajistit zpracování velkého množství dat s jejich následujícím nahráním do paměti PLC. Toto pomohlo, jak zprovoznit stroj, tak i celkově usnadnit a urychlit proces gravírování hliníkových štítků. Byli splněny všechny požadavky firmy a operátorů, včetně těch, co se objevili během testování aplikace za podmínek reálného provozu. Přesto že tento program není ideální, jak z hlediska uživatelského rozhraní, tak i z principů jeho fungování, lze říct, že je to dost spolehlivá aplikace, která je schopná plnit svoje nejdůležitější funkce. Během použití daného programu se vyrostla rychlost produkce vygravírovaných hliníkových štítků a dramatický se zmenšil počet vadných kusů mezi nimi.

SEZNAM POUŽITÉ LITERATURY

- [1] Dot Peen Technology. *Engraving Machines, laser marking systems, dot peen machine* | *SIC Marking* [online]. Pittsburgh: SIC MARKING USA, 2019 [cit. 2019-04-22]. Dostupné z: <https://www.sic-marking.com/dot-peen-engraving>
- [2] Udarno-točejnaja markirovka. *INTERTULMAŠ. Ofitsialnyj sajt. Postavki oborudovanija*. [online]. Moskva: OOO "INTERTULMAŠ", 2018 [cit. 2019-04-22]. Dostupné z: <https://www.itmash.ru/katalog/udarnaya-markirovka-udarno-tochechnaya-oborudovanie-dlya-markirovki>
- [3] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. *Řízení programovatelnými automaty*. Praha: České vysoké učení technické, 1998. ISBN 80-01-01766-4.
- [4] History of the PLC. *Industrial Automation News from AutomationDirect* [online]. Cumming, GA: Library at AutomationDirect.com, 2019 [cit. 2019-05-08]. Dostupné z: <https://library.automationdirect.com/history-of-the-plc/>
- [5] PLC Programmable Logic Controller Hardware Components | PLC Hardware Basics. *Electrical A2Z | Complete Electrical Guide* [online]. WordPress, 2019 [cit. 2019-05-08]. Dostupné z: <http://electricala2z.com/motors-control/plc-programmable-logic-controller-hardware-components-plc-hardware-basics/>
- [6] ISO 8373:2012(en), Robots and robotic devices — Vocabulary. *Online Browsing Platform (OBP)*[online]. Ženeva: International Organization for Standardization, 2012 [cit. 2019-04-23]. Dostupné z: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
- [7] The True Origins of Parallel Robots. *Parallel Robots, Hexapods, Stewart platforms - ParalleMIC* [online]. www.parallemic.org, 2003 [cit. 2019-05-09]. Dostupné z: <http://www.parallemic.org/Reviews/Review007.html>
- [8] Industrial Robot History. *RobotWorx* [online]. Marion: RobotWorx, 2019 [cit. 2019-05-09]. Dostupné z: <https://www.robots.com/articles/industrial-robot-history>
- [9] ÚŘEDNÍČEK, Zdeněk. *Robotika*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2012. ISBN 978-80-7454-223-7.
- [10] Four Layers of TCP/IP model, Comparison and Difference between TCP/IP and OSI models. *Free Networking tutorials, System Administration Tutorials and*

- Security Tutorials - omniseku.com* [online]. www.omniseku.com: OmniSecu.com, 2019 [cit. 2019-05-10]. Dostupné z: <http://www.omniseku.com/tcpip/tcpip-model.php>
- [11] TCP/IP vs. OSI: What's the Difference Between the Two Models?. *Fiberstore | Blog* [online]. New Castle: Fiberstore, 2019 [cit. 2019-05-10]. Dostupné z: <https://community.fs.com/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html>
- [12] What Is the Difference Between Ethernet and Industrial Ethernet?. *Mixed-signal and digital signal processing ICs | Analog Devices* [online]. Norwood: Analog Devices, 2019 [cit. 2019-05-10]. Dostupné z: <https://www.analog.com/en/technical-articles/what-is-the-difference-between-ethernet-and-industrial-ethernet.html>
- [13] Step7 Open Source Ethernet Communication Suite. *Snap7 Homepage* [online]. www.snap7.sourceforge.net/: Snap7 Homepage, 2019 [cit. 2019-05-10]. Dostupné z: <http://snap7.sourceforge.net/>
- [14] Ethernet IEEE 802.3 tutorial. *Electronics Notes: reference site for electronics, radio & wireless* [online]. www.electronics-notes.com: Electronics Notes, 2019 [cit. 2019-05-10]. Dostupné z: <https://www.electronics-notes.com/articles/connectivity/ethernet-ieee-802-3/basics-tutorial.php>
- [15] The Ethernet II Frame Format. *Firewall.cx - Routing Information & Expertise To Network Professionals* [online]. www.firewall.cx: Firewall.cx, 2018 [cit. 2019-05-10]. Dostupné z: <http://www.firewall.cx/networking-topics/ethernet/ethernet-frame-formats/201-ethernet-ii.html>
- [16] 10Base5 - Definition of 10Base5 in The Network Encyclopedia. *The Network Encyclopedia* [online]. Viana do Castelo: Ciberforma Lda, 2019 [cit. 2019-05-10]. Dostupné z: <http://www.thenetworkencyclopedia.com/entry/10base5/>
- [17] Ethernet Wiring. *Practical Networking* [online]. www.practicalnetworking.net/: Practical Networking .net, 2019 [cit. 2019-05-10]. Dostupné z: <https://www.practicalnetworking.net/stand-alone/ethernet-wiring/>

- [18] Internet Protocol. *SearchUnifiedCommunications* [online]. Newton: TechTarget, 2008 [cit. 2019-05-11]. Dostupné z: <https://searchunifiedcommunications.techtarget.com/definition/Internet-Protocol>
- [19] The Transmission Control Protocol. *DePaul University, Chicago* [online]. Chicago: DePaul University, 2019 [cit. 2019-05-11]. Dostupné z: <http://condor.depaul.edu/jkristof/technotes/tcp.html>
- [20] Schéma zapouzdření aplikačních dat na vrstvách TCP/IP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2014 [cit. 2019-05-11]. Dostupné z: https://commons.wikimedia.org/wiki/File:Tcpip_zapouzdeni.svg
- [21] ISO Transport Service on top of the TCP. *IETF Tools* [online]. www.tools.ietf.org: IETF Tools, 2019 [cit. 2019-05-11]. Dostupné z: <https://tools.ietf.org/html/rfc1006>
- [22] S7 Communication (S7comm). *Wireshark Wiki* [online]. <https://wiki.wireshark.org/>: Wireshark Wiki, 2016 [cit. 2019-05-11]. Dostupné z: <https://wiki.wireshark.org/S7comm>
- [23] Overview Of Common Language Infrastructure. *C# Corner* [online]. <https://www.c-sharpcorner.com/>: C# Corner, 2017 [cit. 2019-05-12]. Dostupné z: <https://www.c-sharpcorner.com/blogs/overview-of-common-language-infrastructure>
- [24] .NET Guide. *Microsoft Docs* [online]. Redmond: Microsoft Corporation, 2017 [cit. 2019-05-13]. Dostupné z: <https://docs.microsoft.com/ru-ru/dotnet/standard/>
- [25] C# Guide. *Microsoft Docs* [online]. Redmond: Microsoft Corporation, 2017 [cit. 2019-05-13]. Dostupné z: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PLC	Programmable Logic Controller, Programovatelný Logický Automat
PC	Personal Computer, osobní počítač
GM	General Motors, název společnosti
V/V	Vstup/Výstup
PID	PID regulátor, složený z Proporcionální, Integrovaná a Derivační části
CPU	Central Processing Unit, Centrální Procesorová Jednotka
STL	Statement List, jazyk mnemokódu
LAD	Ladder Diagram, jazyk reléových schémat
FBD	Function Block Diagram, jazyk logických schémat
TCP	Transmission Control Protocol
IP	Internet Protocol
ISO	International Organization for Standardization, Mezinárodní Organizace pro Normalizaci
OSI	Open Systems Interconnection
IE	Industrial Ethernet
UDP	User Datagram Protocol
Gbps	Gigabit za sekundu
DTE	Data Terminal Equipment
DCE	Data Communications Equipment
MAC	Media Access Control
Mbps	Megabit za sekundu
TX	Transmit Data
RX	Receive Data
NIC	Network Interface Card
MDI	Media Dependent Interface

MDI-X	Media Dependent Interface Crossover
PDU	Protocol Data Unit, Protokolová Datová Jednotka
CLI	Common Language Infrastructure
CTS	Common Type System
CLS	Common Language Specification
VES	Virtual Execution System
CLR	Common Language Runtime
FCL	Framework Class Library
CIL	Common Intermediate Language
JIT	Just-In-Time
GC	Garbage Collector
WPF	Windows Presentation Foundation
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language, Rozšířený Značkovací Jazyk

SEZNAM OBRÁZKŮ

<i>Obr. 1. Obecná struktura PLC</i>	14
<i>Obr. 2. Struktury síťových modelů TCP/IP a OSI</i>	19
<i>Obr. 3. Princip přenosu dat pomocí kroucené dvojlinky [17]</i>	22
<i>Obr. 4. Zapouzdření dat v síti TCP/IP [20]</i>	23
<i>Obr. 5. Pořadí protokolů, používaných při komunikaci s PLC společnosti Siemens</i>	24
<i>Obr. 6. Celkový pohled na stroj</i>	29
<i>Obr. 7. Řídící jednotka e10</i>	30
<i>Obr. 8. Rotační stůl a mechanická hlava s hrotem c153</i>	32
<i>Obr. 9. Hliníkový štítek</i>	33
<i>Obr. 10. Označení kabelů pomocí hliníkových štítků</i>	34
<i>Obr. 11. Volba jazyka rozhraní</i>	37
<i>Obr. 12. Hlavní okno programu</i>	37
<i>Obr. 13. Okno nastavení parametrů připojení</i>	38
<i>Obr. 14. Okno s parametry struktury datového bloku</i>	39
<i>Obr. 15. Náhled na okno, obsahující data ve vyrovnávací paměti programu</i>	40
<i>Obr. 16. Okno s daty v paměti PLC</i>	41
<i>Obr. 17. Chyba připojení k PLC</i>	45
<i>Obr. 18. Okno s parametry připojení s hlášením chyby</i>	46
<i>Obr. 19. Chyba při kontrole struktury datového bloku PLC</i>	46
<i>Obr. 20. Příklad chybové hlášky při nahrávání špatného souboru</i>	47
<i>Obr. 21. Chybná úprava vstupních dat operátorem</i>	47
<i>Obr. 22. Detekce neznámé chyby během přenosu dat</i>	48
<i>Obr. 23. Deaktivace částí uživatelského rozhraní</i>	48

SEZNAM PŘÍLOH

P I Obsah vloženého CD-ROM disku

PŘÍLOHA P I: OBSAH VLOŽENÉHO CD-ROM DISKU

Obsah disku:

- Elektronická verze bakalářské práce;
- ZIP-soubor, obsahující zdrojový kód programu ve formě řešení pro vývojové prostředí Visual Studio 2017.