

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA ELEKTROTECHNOLOGIE**



**DIPLOMOVÁ PRÁCE**

**DIPLOMA THESIS**

**AUTOMATIZACE ŘÍZENÍ PROVOZU  
ZKUŠEBNÍ LABORATOŘE**

AUTOR: BC. VÁCLAV SÁNDOR

VEDOUCÍ: ING. PAVEL HRZINA PH.D.

STUDIJNÍ PROGRAM : ELEKTROTECHNIKA, ENERGETIKA A MANAGEMENT

STUDIJNÍ OBOR: TECHNOLOGICKÉ SYSTÉMY

**2019**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sándor** Jméno: **Václav** Osobní číslo: **420059**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra elektrotechnologie**  
Studijní program: **Elektrotechnika, energetika a management**  
Studijní obor: **Technologické systémy**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Automatizace řízení provozu zkušební laboratoře**

Název diplomové práce anglicky:

**Operation Management Automatization of Test Laboratory**

Pokyny pro vypracování:

- 1) Analyzujte potřeby zkušební laboratoře z hlediska automatizace jejího provozu.
- 2) Navrhněte vhodnou strukturu (HW a SW) pro řízení laboratoře s využitím výstupů analýzy a s ohledem na již používaná zařízení.
- 3) Realizujte základní řídicí systém a ověřte jeho funkčnost.
- 4) Zpracujte bezpečnostní analýzu systému a vyhodnoťte stav vámi navrženého systému z hlediska bezpečnosti.

Seznam doporučené literatury:

- [1] Datové listy, návody a dokumentace k systému UNI Pi, dostupné na stránkách <https://www.unipi.technology/cs/>
- [2] ČSN EN ISO/IEC 17025:2018 - Všeobecné požadavky na kompetenci zkušebních a kalibračních laboratoří

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Pavel Hrzina, Ph.D., katedra elektrotechnologie FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **04.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

\_\_\_\_\_  
Ing. Pavel Hrzina, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

V Praze dne 13.5.2019

---

Václav Sándor

## Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Pavlu Hrzinovi PhD., který mi nabídl toto téma a dále mi poskytoval cenné rady při jeho vypracování. Samozřejmě chci poděkovat i za odborný dohled a věcné připomínky.

# Anotace

Cílem mé diplomové práce bylo analyzovat a následně navrhnout řešení automatizace zkušební laboratoře katedry elektrotechnologií. V první části mé práce jsem sbíral a analyzoval požadavky katedry, z nichž jsem posléze vytvořil určitý koncept automatizace. Podle tohoto konceptu jsem také vytipoval potřebné komponenty a navrhl softwarovou část řízení laboratoře. V další části jsem se zaměřil na tvorbu SW a nakonec jsem provedl bezpečnostní analýzu jako po stránce bezpečnosti provozu, tak po stránce „kyberbezpečnosti“.

## Klíčová slova

PLC, HMI, automatizace, bezpečnost, RFID, docházkový systém, sběr dat

# **Annotation**

The goal of this thesis was to analyze and then propose a solution for automation of the testing laboratory of the Department of Electrical Engineering. In the first part of my work I collected and analyzed the requirements of the department, from which I later created a certain concept of automation. According to this concept, I also identified the necessary components and designed the software part of the operation management. In the next part, I focused on SW creation and at the end I carried out a security analysis as in the operation safety as well as in terms of "cybersecurity".

## **Key Words**

PLC,HMI, automation, safety, RFID, attendance systém, data acquisition

# Obsah

Úvod .....	1
<b>Koncepce automatizace laboratoře .....</b>	<b>2</b>
<b>Požadavky Katedry .....</b>	<b>2</b>
<b>1 Vytipování vhodných komponent .....</b>	<b>3</b>
1.1 PLC .....	3
1.1.1 Dostupné možnosti.....	4
1.1.2 UniPi.....	6
1.1.2.1 Programovací prostředí UniPi .....	6
1.1.3 Volba PLC.....	7
1.2 Docházkový systém.....	7
1.2.1 RFID .....	7
1.2.2 Frekvence .....	9
1.2.3 Komunikace.....	9
1.2.4 Řešení .....	10
1.3 Tiskárna štítků .....	10
1.3.1 Řešení .....	11
<b>2 Návrh automatizace.....</b>	<b>12</b>
2.1 Rozvaděč.....	13
2.2 Hardware PLC .....	15
2.3 HMI.....	16
2.4 Schéma a kusovník .....	17
2.5 Nouzové zastavení .....	18
2.6 ČSN EN ISO/IEC 17025.....	19
<b>3 Tvorba programu .....</b>	<b>20</b>
3.1 Programovací jazyky.....	20
3.1.1 LAD.....	21
3.1.2 FBD.....	22
3.1.3 STL.....	22
3.1.4 Další programovací jazyky .....	23

3.2	Datové typy .....	23
3.3	Bloky programu .....	24
3.3.1	OB – Organizační blok.....	24
3.3.2	DB – Datový blok .....	25
3.3.3	FB – Funkční blok.....	25
3.3.4	FC – Funkce .....	25
3.4	Typy paměti.....	26
<b>4</b>	<b>Realizace .....</b>	<b>28</b>
4.1	Konfigurace PLC.....	28
4.1.1	Hardwarová konfigurace .....	29
4.1.2	Konfigurace operačního panelu .....	31
4.2	Funkce automatizace.....	32
4.2.1	Spuštění laboratoře.....	33
4.2.2	Přihlášení do laboratoře .....	35
4.2.3	Docházkový systém .....	37
4.2.4	Funkce tiskárny .....	39
4.2.5	Vedlejší funkce .....	40
4.3	Instalace .....	41
<b>5</b>	<b>Bezpečnostní analýza .....</b>	<b>42</b>
5.1	Bezpečnost laboratoře.....	42
5.2	Zabezpečení laboratoře .....	43
5.3	Připojení do sítě .....	44
5.3.1	Připojení z počítače .....	44
5.4	Kyberbezpečnost .....	46
5.5	Rozdíl oproti stávajícímu řešení .....	47
<b>6</b>	<b>Zhodnocení práce .....</b>	<b>48</b>
6.1	Možná vylepšení.....	48
	<b>Reference .....</b>	<b>49</b>
	<b>Seznam obrázků a tabulek.....</b>	<b>51</b>
	<b>Přílohy .....</b>	<b>53</b>



# Úvod

Diplomová práce bude rozdělena do dvou částí. V první části se budu zabývat koncepcí automatizace laboratoře pro měření baterií, koncepce bude prováděna podle požadavků zákazníka, tedy katedry elektrotechnologie. Nejdříve budu řešit bezpečnost laboratoře, tím je myšlena softwarová část, ale i hardwarová, což znamená, osazení nouzových tlačítek atd.

Po koncepci se přesunu k návrhu zapojení a řešení programu PLC. Hlavním tématem této části tedy bude zapojení komponent v rozvaděči a testování funkčnosti. Dále se přesunu k návrhu úrovní zabezpečení po softwarové stránce vzhledem k požadavkům katedry. Součástí toho bude i návrh docházkového systému, který bude řešen pomocí RFID čtečky, která bude schopna číst karty ISIC a zaměstnanecké čipy.

Další část bude praktická a budu se v ní zabývat realizací projektu. Řízení bude provedeno pomocí PLC, které vytipuji podle potřeby aplikace. Další součástí práce bude návrh rozvaděče, který bude prováděn v programu ProfiCAD. Hlavní důraz bude kladen na funkčnost a bezpečnost. Komponenty rozvaděče budou vytipovány po zjištění požadavků katedry

## Koncepce automatizace laboratoře

Práce bude vyhotovována v souladu s požadavky katedry elektrotechnologie, která laboratoř pro měření baterií obsluhuje. Součástí akreditované laboratoře můžeme chápat i pracoviště pro měření fotovoltaických panelů, které je umístěno ve vedlejší místnosti v budově Fakulty elektrotechnické Českého vysokého učení technického v Praze. Požadavky katedry se tedy budou týkat obou pracovišť a také budou podléhat normám na provoz akreditovaných laboratoří, konkrétně ČSN EN ISO/IEC 17025:2018 - Všeobecné požadavky na kompetenci zkušebních a kalibračních laboratoří.

## Požadavky Katedry

Při konzultaci s vedoucím práce jsme došli k hrubému návrhu automatizace pracoviště, které bude řízeno pomocí PLC s uživatelským rozhraním přes HMI panel. Z důvodu bezpečnosti bude možno pracovat v laboratoři až po přihlášení do systému, které bude řešeno přes panel. Součástí bude docházkový systém, který bude řešen pomocí RFID čtečky, která bude schopna číst ISIC karty a zaměstnanecké čipy, z načtených dat pak bude moci být vytvořen jakýsi zápis práce, kde se po přihlášení pomocí RFID čipu může být uživatel vyzván k vyplnění zápisu práce přes HMI, tzn. úkon, který bude v laboratoři vykonávat, předpokládaná doba atd. Toto vše bude dořešeno až po zprovoznění řízení laboratoře, nicméně se tím v průběhu návrhu bude počítat.

Dalším požadavkem je přidání tiskárny štítků sloužících k identifikaci zakázek laboratoře. Ta může být řízena z PLC a vyplňování textu na štítku bude probíhat přes HMI panel nebo může být řízena z počítače, který je v laboratoři k dispozici. Další možností je připojení do vnitřní sítě, s tím, že do ní bude připojeno PLC i počítač, takže bude možnost tvoření štítků z obou míst.

Všechny tyto body požadavků katedry budou podrobněji rozepsány v rozvaze a návrhu automatizace laboratoře. Po dokončení návrhu bude následovat konečné řešení a implementace řízení.

# 1 Vytipování vhodných komponent

Před začátkem návrhu projektu je třeba vytipovat vhodné komponenty, které budou součástí automatizace. Vhodnost komponent závisí zas a znovu na požadavcích zadavatele projektu. Do vhodnosti se promítá více skutečností, ať závisí na ceně produktů, stávajících možnostech zadavatele nebo potřebě výkonu. Z tohoto důvodu je třeba už od začátku přemýšlet nad správností volby komponent z dostupných možností.

## 1.1 PLC

PLC (Programmable Logic Controller) je průmyslový počítač řízený mikroprocesorem s vlastním operačním systémem, uzpůsobeným pro potřeby řešení automatizačních úloh v reálném čase, s co nejkratší dobou odezvy. Od běžného počítače se liší nejen svým průmyslovým provedením, které je řešeno modulárně, tím pádem je možno automat rozšiřovat různými periferiemi (I/O moduly, sériové karty, řízení pohonů atd.). Dále se liší i svou funkcí, PLC se využívá např. v elektrárnách, na ropných plošinách nebo v jednoúčelových strojí pro výrobu čokoliv. Tím pádem je nutná vysoká spolehlivost, možnost trvalého provozu a schopnost precizního zpracování velkého množství signálů na binární úrovni nebo analogové vstupy. Dále je nutno zvládat převést tyto externí vstupy na interní, vyhodnotit je podle požadavku a vytvořit adekvátní výstupy. [1]

Požadavky na programovatelný logický automat:

- navržen tak, aby byl odolný vůči průmyslovému prostředí, jako je teplota, prašnost, vlhkost, výkyvy napájení.
- schopen zpracovávat běžné úrovně řídicích signálů, digitální signály zpravidla 24Vdc, analogové signály  $\pm 10V / 0-20mA$ .
- jednoduchým způsobem rozšířitelný o další periferie, např. vstupy / výstupy.
- navržen tak, aby umožňovat sledování a diagnostiku řízeného procesu, vyhledávání závad a diagnostiku sebe sama.
- dostatečně rychlý v cyklu zpracování, typicky je doba zpracování do 100ms považována za vyhovující, závisí však od typu procesu.
- navržen tak, aby byla zaručena bezpečnost jak systému, tak zařízení, procesu, technologie.

### 1.1.1 Dostupné možnosti

Největším hráčem na trhu s PLC je bezpochyby Siemens, který sám dává několik možností. Počínaje malými řídicími systémy (LOGO!, SIMATIC S7 řady S7-200 a S7-1200), přes průmyslové automatizační systémy SIMATIC (řada S7-300 a S7-400) a nejnovějšími systémy řady SIMATIC S7-1500. V dnešní době je podle mého názoru Siemens dodavatel na nejvyšší úrovni, a to se také odráží na ceně produktů.

Kdyby se mělo vybírat pouze od Siemensu, tak by úvaha vypadala následovně. Z požadavků na výkon a složitost projektu bychom předem vyloučili řadu malých řídicích systémů jako je právě LOGO! a S7-1200. Malý počet vstupů a výstupů a následná nutnost koupě modulárních periférií také mluví proti této řadě. Navíc uživatelské rozhraní LOGOSOFT!, které není z nejpřívětivějších, ve volbě právě této řady moc nepomůže.

Střední řady S7-300 a S7-400 už by byly lepší volbou. Rozhraní SIMATIC STEP 7 je výborné a samotné PLC je v kompaktní verzi (CPU + I/O periférie) dostatečně vybavené pro danou aplikaci. Osobně bych spíše vybíral z řady S7-300, neboť vyšší řada je určena pro úplně jinou náročnost aplikace (elektrárny, ropné plošiny atd.) a řízení by bylo z hlediska spolehlivosti zbytečně předdimenzováno. Model z řady S7-300 bude určitě mezi hlavními kandidáty na řízení.



Obr. 1 PLC od výrobce Siemens

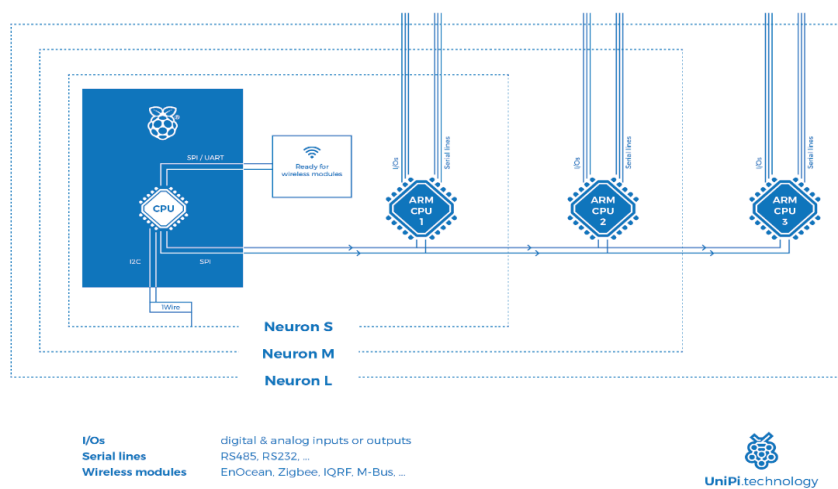
Nejvyšší třída S7-1500 je podobně jako S7-400 kontrolér pro jinou náročnost použití, a právě to se promítá na ceně produktu. Programovací prostředí TIA (Totally Integrated Automation) smazalo některé nedostatky STEP 7, kompletně změnilo vizáž a zaměřilo se více na symbolické adresování než na absolutní, tím pádem je lehčí vytvářet větší projekty. Jenže téměř každý rok vychází nové verze TIA (poslední verzí je TIA V15) a s tím přichází v potaz i koupě dalších licencí, které jsou dražší než u STEP 7.

Jednou z možností je i novinka od brněnských UniPi.technology, kteří své PLC vytvořili z Raspberry Pi, spolu s ním vytvořili i programovací prostředí. Tato možnost je určitě zajímavá, jelikož jde o nové řešení, ale s ním přichází i řada nevýhod. Nicméně se s tímto PLC určitě bude počítat i dál ve volbě vhodného řízení.

S dalšími výrobci PLC jsem přišel do styku pouze minimálně, takže při úvaze volby od těchto dodavatelů nemůžu vycházet z vlastních zkušeností, které jsem dosud získal. Nicméně mezi další výrobce se řadí EATON, Allen-Bradley, Mitsubishi, OMRON atd.

## 1.1.2 UniPi

Po dlouhém přemýšlení jsem zúžil výběr PLC na dva hlavní kandidáty. Prvním z nich je UniPi Neuron L513 od výrobce UniPi.technology sídlícího v Brně. Výrobce postavil celý koncept na minipočítači Raspberry Pi 3 model B, který slouží jako řídicí jednotka samostatných logicky oddělených modulů. Tyto moduly pak mají vlastní procesor a paměť, tudíž jsou schopny pracovat nezávisle na RPi3. Nicméně nemají takový výpočetní výkon a omezená je i konektivita. Soustava modulů a RPi3 je uzavřena v eloxovaném hliníkovém krytu se stupněm krytí IP20. Struktura UniPi je zobrazena na obrázku č. 1. [2]



Obr. 2 Struktura UniPi

### 1.1.2.1 Programovací prostředí UniPi

K ovládání a programování jednotek UniPi Neuron lze využít více programů. Většina je open source, což výrazně přispělo k volbě právě UniPi. Oficiálními softwary jsou však EVOK a Mervis, zatímco EVOK je kompletně opensourcový program, k Mervisu lze přikoupit licenci pro připojení k proxy serveru a ukládání do online databází. Licence však není pro provoz vyžadována, dále je Mervis oproti EVOKu určen pro pokročilejší automatizace a regulace.

Podle výrobce je Mervis vyvíjen a podporuje programování v souladu s normou IEC 61131-3. Mezi nástroje programu patří vývojové prostředí Mervis IDE, grafický editor pro tvorbu ovládacích panelů Mervis HMI a Mervis RT, který umožňuje runtime prostředí pro spouštění a testování programu. [2]

### 1.1.3 Volba PLC

Při výběru jsem se řídil svými zkušenostmi s oběma typy PLC, názory jsou čistě subjektivní. Pro konečný výběr posloužila jednoduchá analýza obou typů.

Siemens S7-300		UniPi L513	
Prověřená technologie	Cena	Cena	Komunikace
Nespočet modulů pro rozšíření	SW vyžaduje licenci	Open source SW	Téměř žádné studijní materiály
Mnoho studijních materiálů a velká komunita	Nutnost 24V zdroje	Má relátkové výstupy, není třeba 24V okruhu	Zatím neprověřená spolehlivost
Mnoho knihoven			
Komunikace			
Intuitivní programovací prostředí			

Tab. 1 Srovnání PLC

Jako vítěz z volby vzešel model od Siemensu z řady S7-300, konkrétní typ bude vybrán během tvorby projektu. Největší nevýhodou Siemensu byla cena, ta však není o mnoho vyšší než cena UniPi a licencovaný software Simatic STEP7, jenže jak jsem zjistil od vedoucího, tak je jedna licence zakoupena i pro školu, takže tím se eliminoval další problém.

## 1.2 Docházkový systém

Další součástí automatizace laboratoře má být docházkový systém. Ten může být elegantně vyřešen využitím stávajících možností (ISIC karet a RFID čipů zaměstnanců) a doplněn systémem přihlašování a popisu činnosti přes HMI panel. V našem případě je samozřejmě zapotřebí vtipovat čtečku o správné frekvenci, dále se musí brát ohled na schopnost komunikace s PLC.

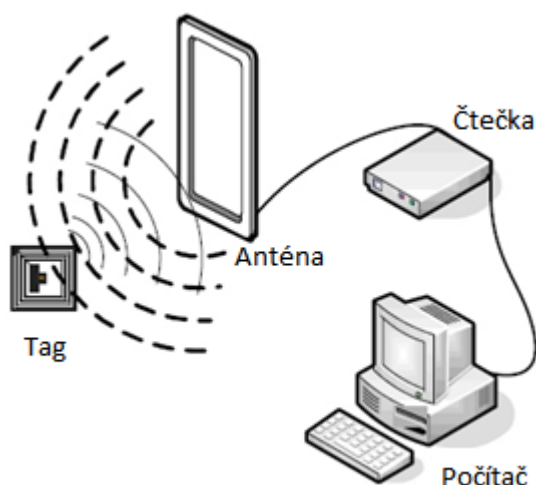
### 1.2.1 RFID

RFID (Radio frequency identification) je technologie, která využívá radiofrekvenční vlny k identifikaci informací, které jsou uloženy na mikročipu. Ten je připojen k anténě a vše je zalito do substrátu (plastu), často se používá polyethylentereftalát (Mylar), který funguje jako ochrana čipu. Tyto komponenty dohromady tvoří tzv. RFID tag, který je ve finálním provedení schopen vyslat uloženou informaci do čtečky, která musí být samozřejmě vybavena anténou. [3]

Čipy RFID tagů se vyrábějí ve třech provedeních, a to read-only (pouze čtení), WORM (jeden zápis, mnoho čtení) a read/write (čtení/zápis).

Dále čipy dělíme:

- **Pasivní** Vysílač (reader) periodicky vysílá pulsy do okolí. Pokud se v blízkosti objeví pasivní RFID čip, využije přijímaný signál k nabití svého napájecího kondenzátoru a odešle odpověď.
- **Aktivní** Používají se méně často než pasivní tagy RFID. Jsou totiž složitější a dražší, jelikož obsahují navíc i zdroj napájení a jsou schopny samy vysílat své identifikace – používají se proto pro aktivní lokalizaci.
- **BAP** systémy (Battery Assisted Passive) Jak již označení napovídá, tento druh čipu využívá obou vlastností předchozích druhů. Při přijímání se chová jako normální pasivní čip, přijímaný signál však využívá k nabíjení přidané baterie, která podporuje sílu vysílaného signálu. [3]



Obr. 3 Soustava RFID



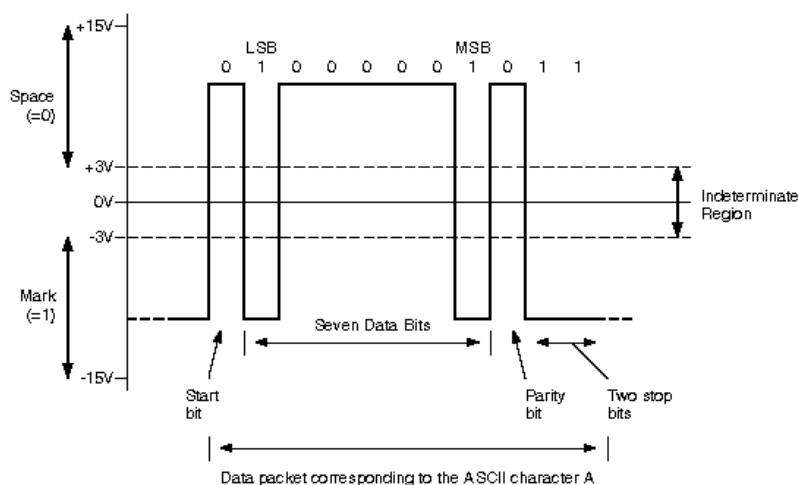
## 1.2.2 Frekvence

V různých e-shopech jsou k dispozici převážně čtečky pracující na frekvencích 125 kHz a 13,56 MHz. Jak jsem již nastínil, je třeba využít stávajících možností, což znamená ISIC karty a zaměstnanecké čipy. Obě možnosti pracují na frekvenci 13,56 MHz, tudíž je nutné se zaměřit na hledání čtečky s tímto parametrem.

- LF 125 kHz
  - krátká čtecí vzdálenost (do cca. 20 cm)
  - nízká přenosová rychlost
  - využití převážně v identifikačních průkazech (evidence docházky), k identifikaci komponent v zařízení během výroby
  - Využívá se pasivních tagů, které se skládají z kotouče měděného drátu a nepřepisovatelné paměti.
- HF 13,56 MHz
  - vyšší čtecí vzdálenost (do cca. 1 metru)
  - nižší rychlost přenosu
  - využití především pasivních tagů
  - nejčastější využití pro knihovní systémy, docházkové systémy, pro identifikační karty (e-peněženky, přístupové systémy)

## 1.2.3 Komunikace

Dalším kritériem je komunikace, kterou je limitována dostupnými možnostmi PLC. Pro komunikaci PLC s RFID čtečkou jsem zvolil sériovou RS232. Standard definuje asynchronní sériovou komunikaci pro přenos dat. Pořadí přenosu datových bitů je od nejméně významného bitu (LSB) po bit nejvýznamnější (MSB). Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se také setkat se 7 nebo 9 bity.



Obr. 4 Popis komunikace RS232

Logický stav „0“, „1“ přenášených dat je reprezentován pomocí dvou možných úrovní napětí, které jsou bipolární a dle zařízení mohou nabývat hodnot  $\pm 5$  V,  $\pm 10$  V,  $\pm 12$  V nebo  $\pm 15$  V. Nejčastěji se používá varianta při které logické hodnotě 1 odpovídá napětí  $-12$  V a logické hodnotě 0 pak  $+12$  V. Základní tři vodiče rozhraní (příjem RxD, vysílání TxD a společná zem GND) jsou doplněny ještě dalšími vodiči sloužícími k řízení přenosu (vstupy DCD, DSR, CTS, RI, výstupy DTR, RTS). Ty mohou a nemusí být používány (zapojeny), nebo mohou být použity pro napájení elektronických obvodů v zařízení, jako je například počítačová myš. [4]

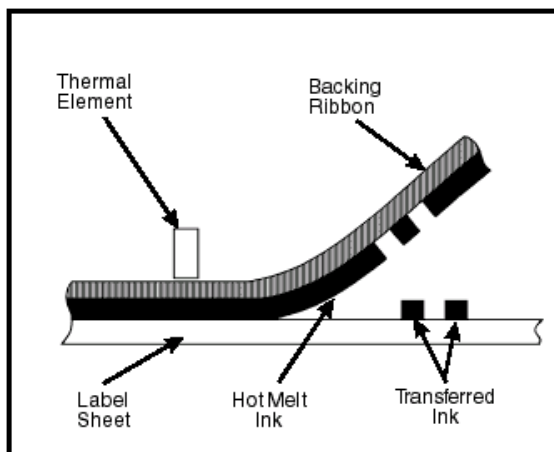
### 1.2.4 Řešení

Čtečku s danými parametry (13,56 MHz, RS-232 a přijatelná cenová relace) jsem hledal na různých e-shopech, bohužel bez výsledku. Nepomohla ani komunikace s největšími dodavateli s tímto druhem zboží. Nejblíže jsem se dostal se čtečkou, kterou mi nabídl technik z firmy SOSelectronic. Nicméně cenová relace těchto čteček je mimo dostupné možnosti, protože musíme počítat i s nákupem jiných, důležitějších komponent.

Po domluvě s vedoucím padlo, že tato čtečka bude diplomovou prací jiného studenta, který bude mít za úkol navrhnout vlastní levné řešení s odpovídající bezpečnostní analýzou.

## 1.3 Tiskárna štítků

Dalším návrhem byla implementace termotransferové tiskárny štítků, která bude využívána hlavně k označování testovaných solárních panelů. Jedná se o nepřímou metodu tištění, kdy se znaky vytisknou nejdříve na karbonovou, voskovou nebo jinou pásku, která se vlivem teploty tiskové hlavy nahřeje a následně prudce schladí. Tím se uhlík, vosk nebo jiný materiál přenesou na výrobek. V kontrastu s termotransferem obyčejný „termotisk“, což je přímý způsob, nelze v průmyslu použít, protože se jím nedá v dobré míře značit jiné materiály než papír. Zde se také používá teplo, ale není při tisku přítomna žádná páska. Je však třeba speciálního papíru (nejčastěji termochromického), kdy vzor na papíru ohřeje a tím se na papíru zbarví oblasti, které vzor tvoří.



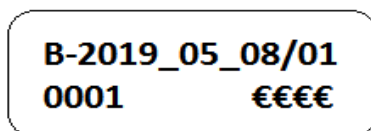
Obr. 5 Termotransferový tisk

Komunikace bude řešena nejspíš pomocí sériové komunikace RS-232, aby mohla komunikovat s PLC. Hlavním parametrem, mimo ty technické, bude cena, která vzhledem k počtu testovaných panelů za rok nesmí být příliš vysoká. Nemá totiž smysl kupovat průmyslovou tiskárnu, které jsou dimenzovány na produkci štítků v řádu tisíců denně, když naše aplikace vyžaduje tisk maximálně několik stovek ročně. [5]

### 1.3.1 Řešení

Po průzkumu dostupných možností jsem zvolil termotransferovou tiskárnu Zebra GK420T od společnosti Zebra. Tato společnost je známá svým působením v odvětví průmyslového tisku, ba dokonce vytvořila svůj programovací jazyk ZPL pro vytváření štítků. Příkazy psané tímto jazykem jsou pak posílány přes danou komunikaci do tiskárny. Já tedy využiji sériovou komunikaci RS232.

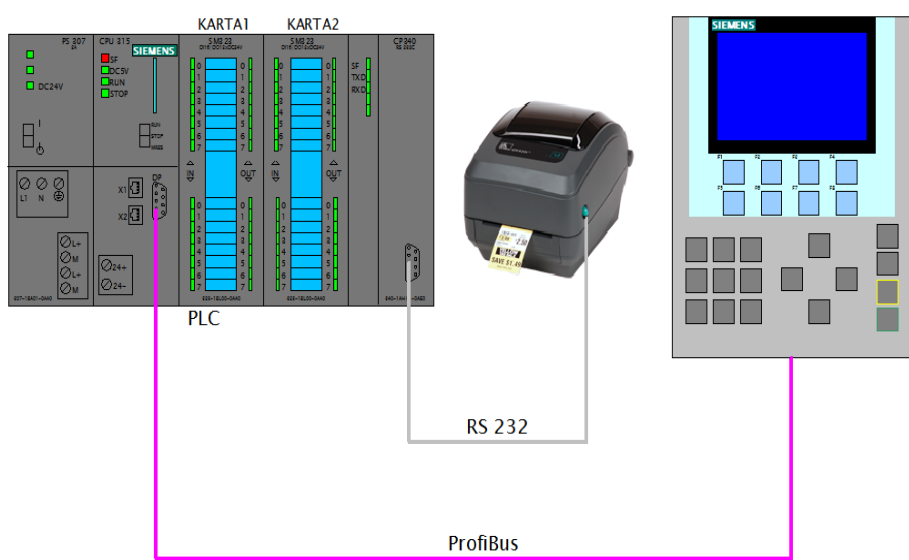
Vzhled štítku by měl být následující. Měl by obsahovat druh měření, datum vystavení zakázky, číslo a libovolně může obsahovat vlastní označení nebo rozlišovací znaky podle toho, jestli je daná zakázka od zákazníka nebo vnitřní z ČVUT.



Obr. 6 Příklad štítku

## 2 Návrh automatizace

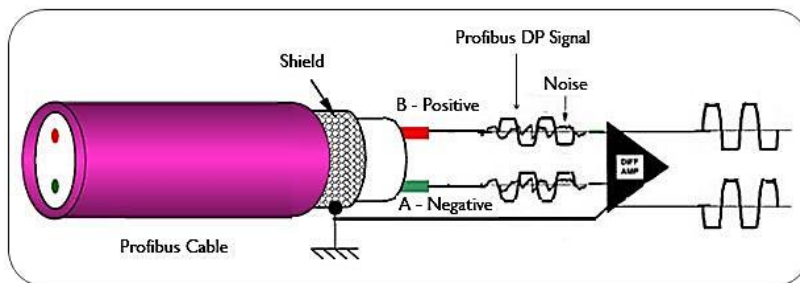
Po úvaze nad klady a zápory vybraných programovacích automatů jsem se rozhodl pro Siemens řady S7-300, i přes nutnost 24V zdroje pro ovládací okruh. Zjednodušené schéma zapojení je jednoduché, PLC bude s HMI komunikovat přes ProfiBus a s tiskárnou po sériové RS-232. ProfiBus je nastaven na přenosovou rychlost 1,5MB/s s nejvyšší ProfiBus adresou 126.



Obr. 7 Schéma řízení

Profibus je průmyslovou sběrnici, která nachází v automatizační technice široké uplatnění, a to jak v oblasti průmyslové automatizace - Profibus DP, tak i v oblasti procesní automatizace – Profibus PA. V roce 1996 byla přijata sběrnice PROFIBUS jako evropská norma EN50170, což zaručuje její standardizaci.

Profibus DP je z Profibusů nejpoužívanější. Slouží pro rychlou cyklickou komunikaci mezi PLC a jejich decentralizovanými vstupy a výstupy. Síť může být typu monomaster nebo multimaster. Při konfiguraci sítě monomaster je řízení typu master/slave, při multimaster si jednotlivé řídicí jednotky umožňují přístup na sběrnici metodou token passing, a komunikace mezi řídicí jednotkou, která vlastní token a podřízenou je opět master/slave. Profibus DP je vybaven diagnostickými funkcemi pro monitorování stavu systému z hlediska spolehlivosti a bezpečnosti. Na jeden segment sběrnice sítě je možno připojit až 32 zařízení (aktivních/pasivních) a pro připojení více zařízení je nutno použít opakovače. [6]



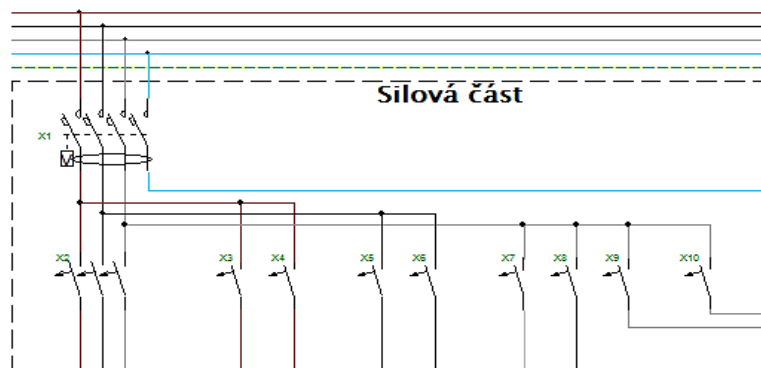
Obr. 8 Popis kabelu ProfiBusu

Přenosová rychlost pro Profibus PA je konstantní 31,25 kb/s (dáno standardem IEC 1158-2). Doba reakce u Profibus DP pro přenosové rychlosti 500 kb/s, 1,5 Mb/s a 12 Mb/s, znázorněná v grafu, je přijatelná pro klasické úlohy řízení v reálném čase (např. pro přenosovou rychlost 1,5 Mb/s, 1 master a 10 slave je informace o vstupech a výstupech přenesena mezi zařízeními master a slave během 3 ms).

Další výhodou pro řízení v reálném čase je fakt, že přístup na sběrnici je deterministický, tudíž lze garantovat cyklus sběrnice (dobu oběhu tokenu). Díky těmto vlastnostem se Profibus DP začíná používat i pro řízení servomechanismů v obráběcích strojích (ProfiDrive Profil). Obecně lze říci, že komunikace po Profibusu DP neomezuje scan cyklus PLC, jenž je limitován mimo jiné dobou výpočtu řídicího algoritmu. [6]

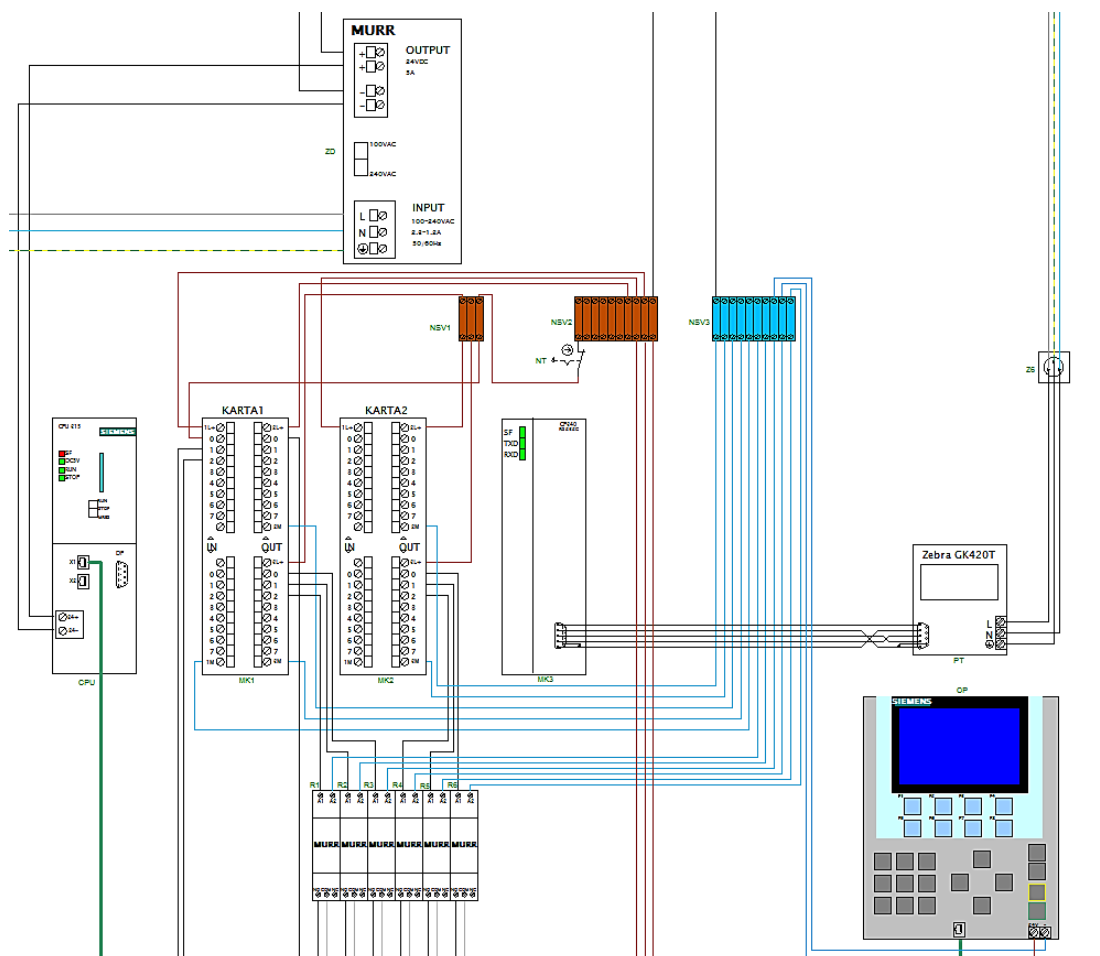
## 2.1 Rozvaděč

Napájení laboratoře je samozřejmě přivedeno ze školní třífázové sítě přes proudový chránič PF6-25/4/003 od Eatonu. K jištění byly vybrány jističe od Noarku, od stejného výrobce se pak vybraly 230V stykače, které budou sloužit ke spínání zásuvek.



Obr. 9 Přívod ze sítě a ochranné prvky

Jelikož jsem jako řídicí jednotku vybral PLC od Siemensu, které je napájené 24 V, bylo nutné pořídit i 24V zdroj. Jako vhodnou variantu jsem vybral jednofázový, spínaný ECO-Rail 2 od MURR Electronics, který je schopen dodávat až 5 A, napájet bude CPU, HMI i výstupy PLC, které jsou také 24V.

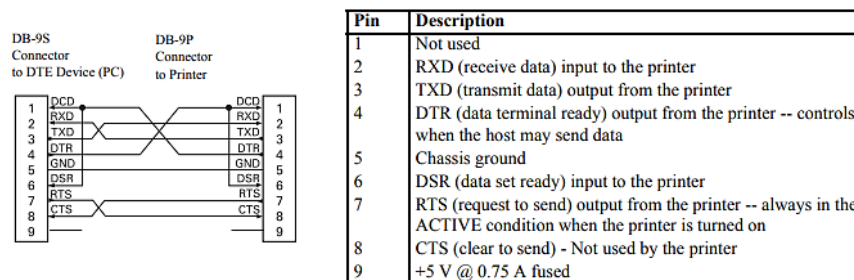


Obr. 10 Zapojení řídicí části

Výstupy PLC povedou na cívku 24V relé od MURRu, ty pak budou spínat fázi, která vede na cívku 230V stykač od Noarku. Důvod tohoto zapojení je prostý, stykače od Noarku již byly k dispozici, proto jsem je využil. Dokoupit pár 24V relétek bylo výhodnější, než kupovat nové, o řád dražší 24V stykače. Tím pádem je řídicí část galvanicky oddělená od silové.

Tiskárna je pak propojena se sériovou kartou PLC přes RS-232. Když vybereme handshaking XON/XOFF, datový to pak řídíme pomocí řídicích příkazů DC1 (XON) a DC3 (XOFF). Signál z DTR (Data terminal ready) je pro nás pak nedůležitý.

Modulační rychlost je nastavena na 9600 baud (bit/s), délka wordu je 8 bitů, žádná parita, 1 bit je nastaven jako STOP bit a handshaking jsem vybral XON/XOFF. Komunikace je vedena po devítižilovém kabelu s koncovkami Cannon DB-9 (samice u tiskárny i PLC). Zapojení kabelu je zobrazeno na obrázku č. 11.



Obr. 11 Zapojení kabelu Cannon DB-9 k tiskárně

## 2.2 Hardware PLC

Siemens všeobecně vsází na modularitu svých výrobků v oblasti programovacích logických automatů, tím pádem si zákazník může sestavit řídicí jednotku, jak potřebuje.

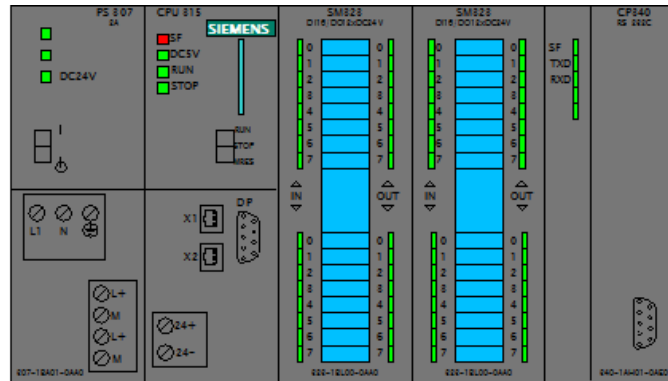
Jako CPU jsem zvolil Siemens CPU 315-2 DP s integrovanou pamětí 256 kB a další 128 kB pro retentive bloky (bloky, které uchovají informaci i po odpojení napájení). Pro komunikace lze využít MPI nebo ProfiBus, právě ten bude využit pro komunikaci s HMI panelem. Toto CPU má relativně krátký čas zpracování, jak je zobrazeno v tabulce níže.

CPU processing times	
for bit operations, typ.	0.05 $\mu$ s
for word operations, typ.	0.09 $\mu$ s
for fixed point arithmetic, typ.	0.12 $\mu$ s
for floating point arithmetic, typ.	0.45 $\mu$ s

Tab. 2 Cycle time CPU pro různé operace

Za CPU jsou zařazené dvě I/O karty typu SM323, které mají po šestnácti vstupech a po šestnácti výstupech. Vstupní WORD (16 bitů) má jedno napájení a každý výstupní BYTE (8 bitů) musí mít své napájení. [11]

Poslední kartou PLC je sériová komunikace CP340, která zajišťuje posílání dat protokolem RS-232C do tiskárny. Ta je vybavena konektorem Cannon DB-9 a funguje jako vysílač i přijímač dat. Sestavené PLC je zobrazeno na obrázku č. 12.



Obr. 12 Sestavené PLC

## 2.3 HMI

Laboratoř je také nutné ovládat, k tomu využijí ovládacího panelu KP400 Comfort také od Siemensu. KP v názvu vypovídá o nedotýkové obrazovce, 400 odkazuje na čtyřpalcový displej a přídomek Comfort značí, že panel má svou klávesnici. Komunikace může probíhat po ProfiNetu, MPI nebo ProfiBusu. Velkou výhodou nové generace panelů je, že dokáže vytvořit http server, na který se dá připojit pro stažení dat. Tento atribut využijí pro logování docházkového systému, který pak budu stahovat přímo z 4MB karty, která je k panelu připojena.

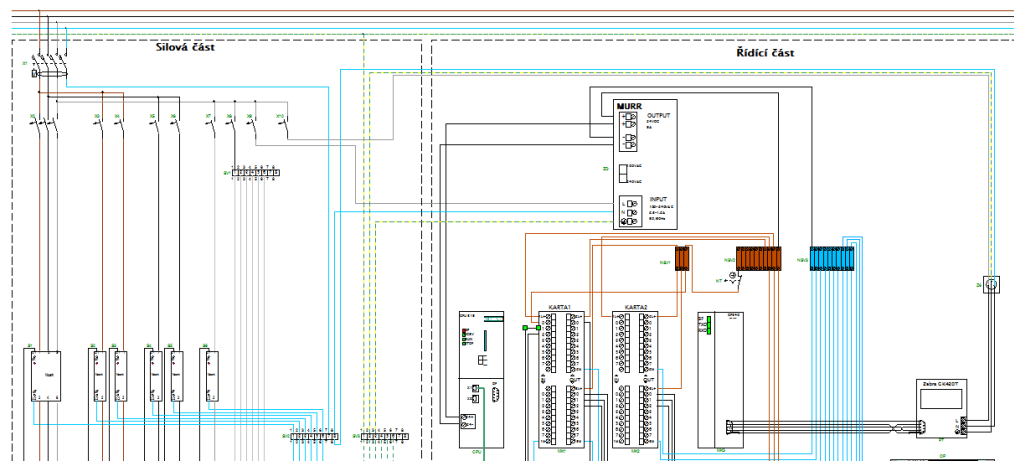


Obr. 13 Panel KP400 Comfort



## 2.4 Schéma a kusovník

Schéma bylo vytvářeno v programu ProfiCAD a je součástí této práce. Celé schéma je rozděleno na dvě části, řídicí a silovou. Řídicí je od silové oddělena pomocí 24V relé a obsahuje PLC, HMI a ostatní 24V prvky.



Obr. 14 Ukázka schématu

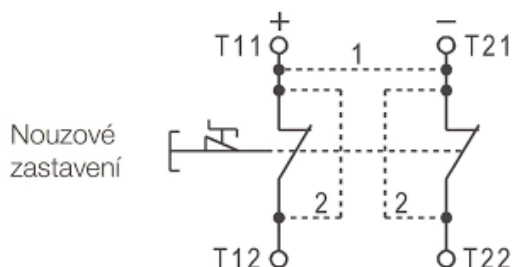
Výhodou programu je i generování kusovníku a umístění komponent ve výkresu. Nicméně jsem generovaný kusovník ještě poupravil a znovu rozdělil součástky do silové a řídicí části. Umístění každé součástky je podrobně rozepsáno v konkrétním rozpisu součástek.

Kusovník laboratoř			
Výrobce	Název	Typ	Počet kusů
Noark	PNS 48T	Rozvodnice s krytím IP40	1
EATON	PF6-25/4/003	Proudový chránič s jištěním 25A	1
Noark	Ex9BN 100053	3f jistič (B16)	1
Noark	Ex9BN 100008	1f jistič (B16)	8
Noark	Ex9BN 100004	1f jistič (B6)	1
Eleman	1000283 HLAK 25-5/10	Svorkovnice 230V	3
Noark	Ex9CH2540	3f stykač ovl. 230V	1
Noark	Ex9CH2510	1f stykač ovl. 230V	5
CEE	Socket 32A 432RS6	Zásuvka 3f	1
ABB	5517-2389 B1	Zásuvka 1f	6
MURR Electronic	ECO-Rail 2	Spínaný zdroj, 1fázový	1
Siemens	CPU 315-2 PN/DP	CPU PLC	1
Siemens	SM323	Karta DI16/DO16	2

Tab. 3 Ukázka kusovníku

## 2.5 Nouzové zastavení

"Nouzové zastavení" znamená, že v případě nouze je nutno nejen zastavit všechny nebezpečné pohyby, ale i bezpečně vyřadit všechny zdroje energie, které mohou být zdrojem nebezpečí, jako například nahromaděná energie.



Obr. 15 Schéma nouzového tlačítka

Základní norma pro bezpečnost strojních zařízení ČSN EN 60204-1 ed.2 zjednodušeně řečeno říká, že každý stroj musí být vybaven nejméně jedním tlačítkem nouzového zastavení se stop funkcí (funkce zastavení) kategorie 0 (tj. okamžité vypnutí odpojením napájení do pohonů stroje). Stop funkce kategorie 1 (řízené vypnutí, kdy k vypnutí energie dojde teprve po dosažení klidového stavu stroje) nebo kategorie 2 (řízené vypnutí, kdy dodávka napájení do pohonů stroje je zachována) musí být zajištěny pouze v případě potřeby, vyžadují-li to technickobezpečnostní a/nebo funkční požadavky stroje.

Nouzové zastavení pak podléhá následujícím pravidlům:

- nouzové zastavení musí být snadno přístupné,
- nouzové zastavení musí nebezpečný stav ukončit co možná nejrychleji, aniž by vznikla dodatečná rizika,
- povel "Nouzové zastavení" musí mít přednost před všemi ostatními funkcemi a povely ve všech provozních režimech,
- reset "Nouzového zastavení" nesmí mít za následek opětovné spuštění stroje, musí být použit princip přímého stisknutí s mechanickou blokovací funkcí.

Obyčejně se v automatizaci využívá bezpečnostních relé, které mají dva bezpečnostní kanály. Pokud se jedná o výrobní stroj, jedním kanálem je právě zpravidla několik STOP tlačítek zapojených v sérii a na druhém kanále jsou zase v sérii zapojené bezpečnostní magnety nebo koncové spínače, které jsou připevněny na krytech. Nicméně v našem případě je možno využít pouze jednoho kanálu, protože není třeba krytů. [7]

Tlačítko nouzového zastavení představuje absolutní základ alespoň pro základní bezpečnost strojů. Sice samo o sobě nijak uživatele a obsluhu zařízení nechrání, ale umožňuje včasnou aktivaci významně snížit následky nebezpečné situace. V jeho případě je tak hlavním parametrem jednotný a všeobecně známý vzhled a spolehlivost funkce.

Tudíž padla otázka, jestli odepínat nouzovým zastavením pouze řídicí fázi stykačů nebo najít jiný způsob. Rozhodl jsem se, že zastavením budu odepínat pouze výstupní kartu PLC. Tím pádem budu mít od nouzového tlačítka signál, že je vše v pořádku nebo naopak, že je stisknuto. S tím se pak dá dále pracovat v programu a v ovládacím panelu.

## 2.6 ČSN EN ISO/IEC 17025

Celý návrh automatizace také musí podléhat normě ČSN EN ISO/IEC 17025. Pod tímto označením, dále jen 17025, se ukrývá norma s celým názvem „*Posuzování shody – Všeobecné požadavky na způsobilost zkušebních a kalibračních laboratoří*“. Zkratka ČSN v tomto případě značí, že se jedná o českou verzi evropské normy EN ISO/IEC 17025. ISO/IEC označuje, že norma vznikla v součinnosti dvou organizací. Tou první je mezinárodní organizace pro standardizaci, zkráceně ISO, a druhou organizací je mezinárodní elektrotechnická komise, zkráceně IEC.

Z hlediska historického vývoje normy 17025 se v České republice pro posuzování shody uplatňovaly požadavky dle normy EN 450013 a ISO/IEC Guide 254. V roce 2000 byly standardy ISO nahrazeny novou verzí ISO 9001:2001. Tato revize vedla k úpravě normy 17025:2001 a vyšla nová verze 17025:2008. V roce 2008 byla provedena další úprava ISO 9001 a následně také v roce 2016. Na tyto změny nyní znovu reagovala norma 17025 a po velmi dlouhém období byla přijata nová verze 17025:2018.

Jak je patrné z předchozího odstavce, norma 17025 je velmi úzce spjata se standardem ISO 9001. Standard ISO 9001 se zabývá požadavky na systém managementu kvality. Těmito požadavky se zabývá i norma 17025, avšak navíc řeší požadavky na technickou způsobilost zkušebních a kalibračních laboratoří. V jistém zjednodušení ji tedy lze chápat jako specifický případ aplikace standardu ISO 9001, rozšířený o odbornou část. [8]

## 3 Tvorba programu

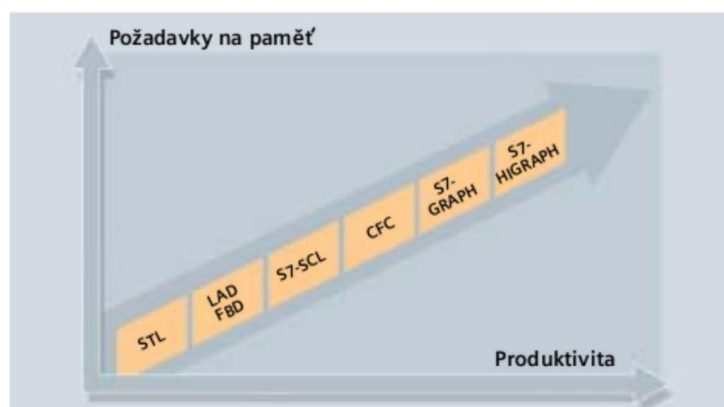
Jak již bylo nastíněno, programovat se bude ve dvou různých programech od stejného dodavatele, a to Simatic STEP S7 a jeho pomyslném nástupci TIA Portalu od vývojáře Siemens. Hlavní rozdíl mezi těmi to dvěma programy je orientace adresování proměnných a dat. V Simaticu se kladl důraz hlavně na absolutní adresování, kdy nebylo například možné databáze optimalizovat do minimální velikosti. Jedním deklarovaným bitem mohl člověk při nešetrném adresování zabrat celý doubleword (32 bitů). Tento nešvar vymýtilo symbolické adresování, kdy si TIA Portal deklaruje adresy hlavně podle symboliky, absolutně až na druhém místě.

Také se sjednotilo programovací prostředí pro PLC a HMI. Dříve se ve STEP 7 vytvářel program PLC a HMI až v nadstavbě WinCC, kdy každý editor vypadal jinak, a vytváření projektů bylo o něco složitější, než je tomu dnes v TIA Portalu.

### 3.1 Programovací jazyky

I přes určité rozdíly v těchto dvou programech, jedno zůstává stejné, a to programovací jazyky. Program se dá vytvářet v sedmi různých jazycích, některé jsou pro určité aplikace vhodnější než jiné. Také existuje určitá závislost mezi požadavkem na paměť a produktivitou programování.

Například programování S7-GRAPH a S7-HIGRPAH je sice efektivní, ale ne vždy se vyplatí zde začít programovat, jelikož se jedná jen o cyklické operace. Velmi často se stává, že program napsaný v tomto prostředí se stane téměř nečitelným, a to i pro tvůrce programu z důvodu velkého počtu ošetření mezistavů.

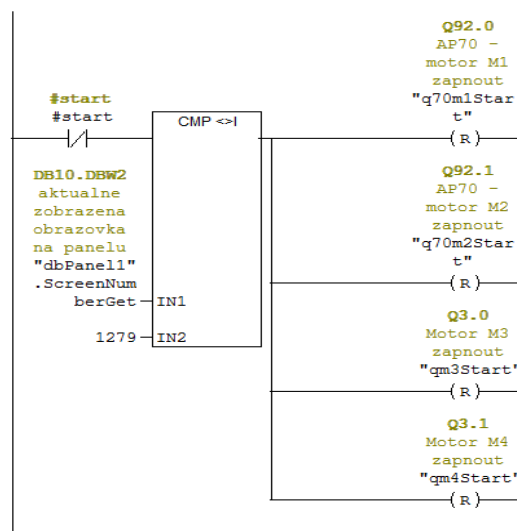


Obr. 16 Srovnání jazyků v závislosti na produktivitě a požadavku na paměť

Programovací jazyky využívané u PLC kontrolérů jsou definované mezinárodní normou IEC 61131-3. Tato norma definuje kromě syntaxe těchto jazyků také základy architektury PLC systémů. Mezi hlavními třemi, tedy LAD, STL a FBD lze obvykle přepínat, u ostatních už záleží na obsahu kódu. Celý projekt bude vytvářen převážně v STL, poněvadž se s ním lépe pracuje s daty.

### 3.1.1 LAD

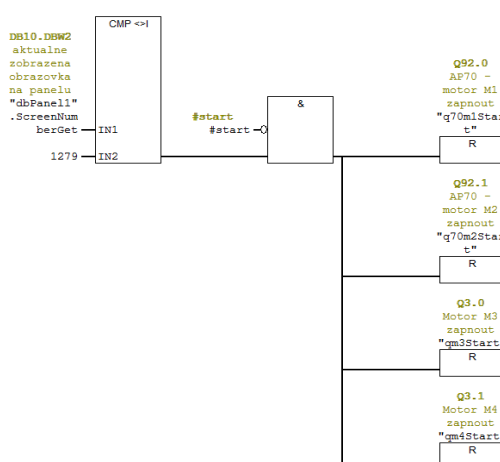
LAD neboli žebříčkové schéma patří k nejstarším metodám programování PLC. Mezi základní elementy patří kontakty typu NO (Normally Open), NC (Normally Closed), vstupní relé, časovače a čítače. K ovládání ladder diagramů se využívá kombinační a sekvenční logika. Kombinační logika obsahuje sériové nebo paralelní zapojení elementů LAD, pomocí kterých je realizována logika programu. Sekvenční logika je využívána v případě nutnosti zpětné vazby, tzn.: výstup jedné části programu je vstupem pro jinou část programu. [9]



Obr. 17 Ukázka jazyku LAD

### 3.1.2 FBD

FBD neboli Function Block Diagram je také graficky orientovaný jazyk, avšak tentokrát v podobě bloků. Patří k nejrozšířenějším možnostem programování PLC. Základem FBD je funkční blok, čímž rozumíme instrukce, které na základě stavu vstupů nastaví odpovídající stavy výstupů. Nejzákladnější bloky využívají výrokovou logiku. Proto se zde setkáme s bloky AND, OR, NOT, XOR, NAND, NOR, přiřazení hodnoty atd. Na následujícím obrázku je stejný kus kódu jako bylo u ladderu, akorát přepnutý do FBD. [10]



Obr. 18 Ukázka jazyku FBD

### 3.1.3 STL

Statement List je nižší programovací jazyk připomínající Assembler, který pomocí volitelného formátu nabízí určitý stupeň flexibility při programování a orientaci v programu. To však vyžaduje jisté zkušenosti, STL je často používané zkušenými programátory. Umožňuje řešit i problémy, které se nedají řešit pomocí editorů LAD a FBD. Například složitější matematické operace nebo ukládání a čtení z akumulátorů. Editor STL používá pouze instrukční soubory SIMATICU. Vykonání instrukcí je v této formě zobrazení nejrychlejší.

```

AN      #start                #start
A(
L       "dbPanel1".ScreenNumberGet  DB10.DBW2
L       1279
<>I
)
R       "q70m1Start"          Q92.0
R       "q70m2Start"          Q92.1
R       "qm3Start"            Q3.0
R       "qm4Start"            Q3.1

```

Obr. 19 Ukázka jazyku STL

### 3.1.4 Další programovací jazyky

**SCL (STRUCTURED CONTROL LANGUAGE)** – vyšší programovací jazyk založený na Pascalu. Tento jazyk bude využit k programování sběru dat.

**GRAPH** – jazyk velmi připomíná vývojový diagram programu, hodí se na sekvenční programování.

**HiGRAPH** – jedná o vylepšenou verzi GRAPHU, tato verze připomíná ještě více stavový diagram.

**CFC (CONTINUS FUNCTION CHART)** – umožňuje technologické požadavky, které mají být transformovány do spustitelných programů. Je možné nejdříve definovat bloky, připojit je, a až poté parametrizovat.

## 3.2 Datové typy

Datové typy základně rozdělujeme na základní a komplexní. Elementární, tedy základy datové typy definují strukturu datových prvků, které nemohou být rozděleny na menší jednotky a odpovídají definici uvedené v normě DIN N 1131-3. Základní datový typ definuje paměťovou oblast, pevnou velikost a představuje bit, integer, real, období, čas a charakter hodnoty. Následující tabulky obsahují popis datových formátů. První tabulka popisuje základní datové formáty, druhá tabulka popisuje komplexní datové formáty.

Typ	Klíčové slovo	Bitová šířka	Rozsah
Bit	BOOL	1	TRUE, FALSE
Byte	BYTE	8	Různé, záleží na použití
Word	WORD	16	
Double word	DWORD	32	
Individual character	CHAR	8	ASCII
Integer	INT	16	-32 786 až 32 767
Double integer	DINT	32	-2 147 483 648 až 2 147 483 647
Floating point number	REAL	32	-3,4028522E+38 až +1,17549E-38
S5 time	S5TIME	16	0H 0M 0S 10MS až 2H 46M 30S
Date	DATE	16	1990-01-01
Time of day	TIME_OF_DAY	32	0:0:0 až 23:59:59,999

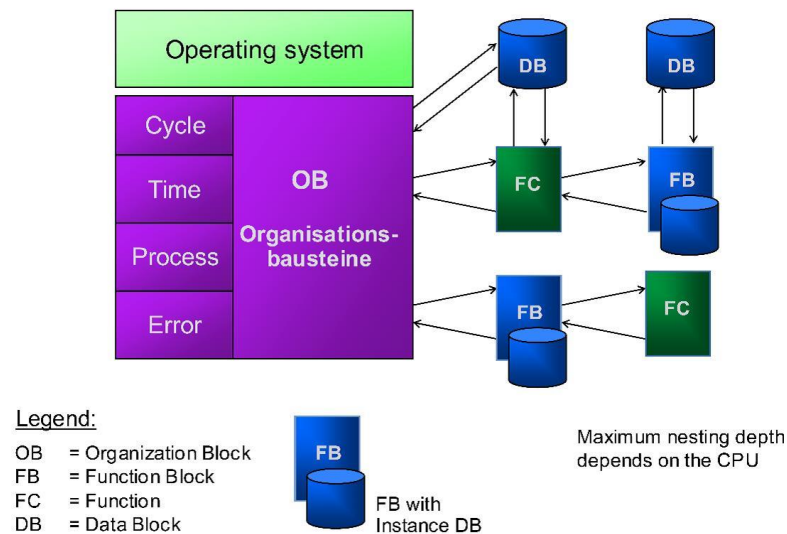
Tab. 4 Základní datové typy

Datový typ	Popis
DATE_AND_TIME_DT	Šířka 64 bit, ukládá čas a datum
STRING	Řetězec dat, až 256 znaků
ARRAY	Pole dat
STRUCT	Definuje skupinu datových typů v libovolné kombinaci

Tab. 5 Komplexní datové typy

### 3.3 Bloky programu

Program se dělí do bloků, které jsou nadále rozděleny podle jejich funkcí. V popisu bloků je čerpáno z učebních materiálů pro programování v TIA Portalu.



Obr. 20 Struktura programu PLC

#### 3.3.1 OB – Organizační blok

Je základním blokem v programu. Při vytvoření projektu se automaticky vytvoří i blok Main (OB1). OB jsou volány cyklicky, odkud se provádí volání bloků a dochází k jejich zpracování. Organizačních bloků existuje několik druhů:

- OB Startup, který se vykoná jen jednou při uvedení PLC do provozu z režimu STOP.
- OB, které jsou cyklicky volány pro vykonávání programu.
- OB pro zpracování alarmů a chyb.



### 3.3.2 DB – Datový blok

Slouží k uložení dat programu. Datové bloky obsahují hodnoty proměnných, které jsou využívány v programu. Každý funkční blok v programu může do DB zapisovat data nebo z něj číst.

### 3.3.3 FB – Funkční blok

Funkčními bloky jsou tvořeny podprogramy, které jsou vždy vykonány, když je funkční blok volán. Takovéto volání se nazývá instance funkčního bloku. Pro každé volání funkčního bloku je přidělen datový blok, do kterého se ukládají zpracovaná data a parametry funkčního bloku. Informace obsažené v tomto datovém bloku jsou uchovány i po vykonání FB.

Funkční bloky se používají tehdy, když pro vykonání úlohy nestačí funkce, jinými slovy, je potřeba uchovat informace i po skončení volání funkčního bloku, pro další běh programu, které se uchovají v přiděleném datovém bloku.

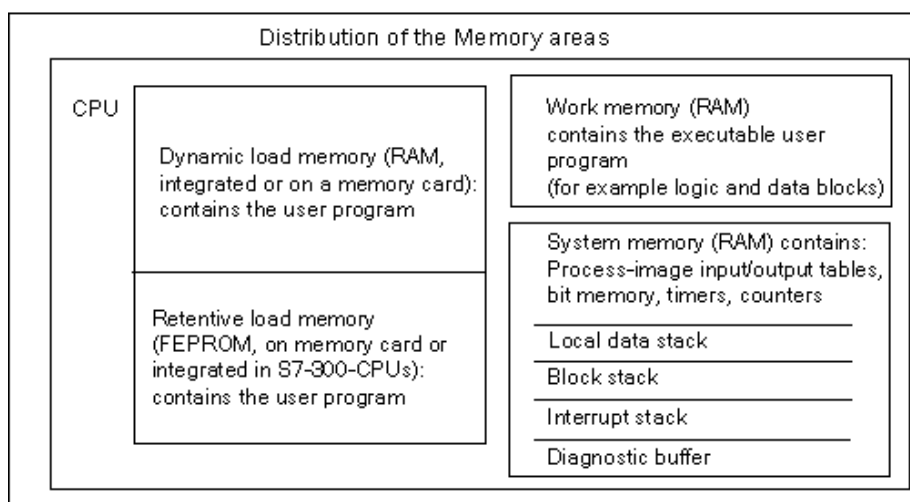
### 3.3.4 FC – Funkce

Funkce obsahuje program, který se vykoná po volání z jiného bloku. K funkci se neváže žádná přidělená paměť, po vykonání funkce jsou použítá data ztracena. Funkci je možné volat několikrát z různých částí programu. Proto je funkce vhodná pro napsání jednoduchých částí programu, které se často opakují.

### 3.4 Typy paměti

Paměť S7 CPU se dá rozdělit do tří částí. První je tzv. Load memory, která slouží k uložení programu, nicméně tento nahraný program je uložen bez symbolických adres a komentářů. Z této paměti lze i číst a to při porovnávání programu v PLC s programem v počítači nebo při uploadu nahraného programu. Tato paměť může být RAM nebo EEPROM.

Další paměť je integrovaná RAM, která obsahuje části S7 programu, které jsou nezbytné pro funkci. Nazývá se také pracovní oblastí. Poslední z částí paměti je RAM, která obsahuje prvky paměti, které poskytuje každé CPU pro správný běh programu. Jsou jimi vstupy a výstupy, paměťové bity (merkery), časovače a country.



Obr. 21 Rozdělení paměti PLC

Systémovou paměť neboli RAM ještě dělíme do dalších oblastí, na které pak v programu pomocí absolutního adresování odkazujeme. Ke každé oblasti pak můžeme přistupovat pomocí různých datových typů, od základních bitů (dvoustavová logika) až po doublewordy (32 bitů). [11]

První z oblastí jsou vstupy, na které se odkazuje pomocí písmena I. Vstupní bity jsou pak označeny např. I0.0 a doublewordy ID0. Podobně se v programu odkazujeme i na ostatní oblasti. Na začátku každého skenu CPU vyhodnotí vstupy ze vstupních modulů a uloží je do této paměti. U výstupů je proces opačný, v průběhu cyklu se vyhodnocují a vypočítávají hodnoty výstupů, ty se pak uloží do oblasti, na kterou se odkazuje pomocí písmena Q. Na konci cyklu pak pošle vyhodnocené hodnoty na dané výstupové moduly.

Dalšími oblastmi jsou timery (T), countery (C) a tzv. merkery (M). Timery slouží jako paměť pro časovače CPU. Countery pak pro čítače a merkery pro ukládání mezivýsledků v programu. Velikosti těchto oblastí má každé CPU různé.

Následují data z datových bloků (DB), které fungují stejně jako merkery. Data z datových bloků jsou však na rozdíl od M-paměti inherentně retentive (hodnoty jsou uloženy i po odpojení napájení), zatímco u merkerů je nutné tuto vlastnost nastavit v hardwarové konfiguraci.

Posledními z oblastí jsou lokální (L) a periferní vstupy a výstupy (PI/PO). Lokální slouží jako paměť pro určitý blok, zatímco se vykonává jeho cyklus. Periferní I/O umožňují přístup k centrálním a vedlejším I/O modulům.

## 4 Realizace

V první části realizace se zaměřím celkově na programovou část. Program PLC bude vytvářen v prostředí Simatic STEP 7 a grafické prostředí HMI bude vytvářeno v TIA Portal V13. Program v PLC lze také vytvářet v TIA Portal, nicméně tento software škola nemá k dispozici, a proto bude lepší pro budoucí údržbu využít k programování PLC software, který mají licencovaný. Sběr dat z docházky bude řešen v TIA Portalu a ukládání bude probíhat na kartu připojenou k ovládacímu panelu.

V další části dojde hardwarové realizaci, kdy se vybaví rozvaděč příslušnými přístroji, připojí se do rozvodné sítě a bude se zkoušet funkčnost řešení automatizace laboratoře se všemi požadavky.

### 4.1 Konfigurace PLC

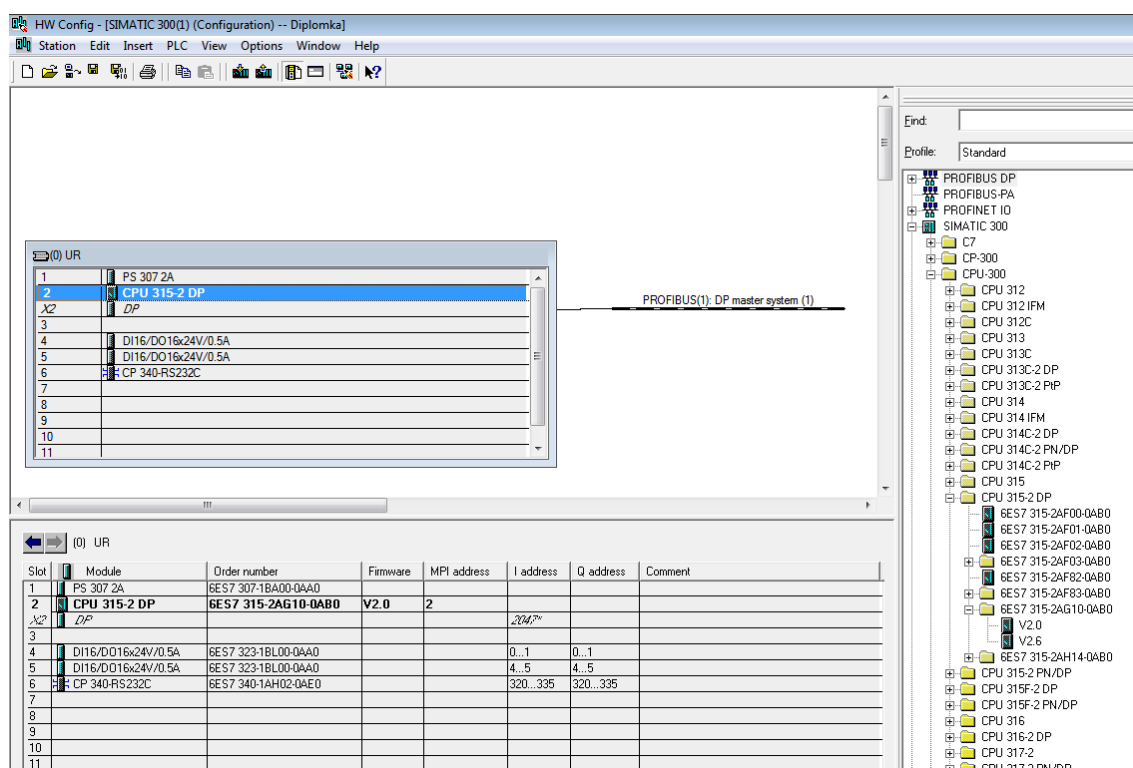
Funkce programu byly probrány předem ve fázi koncepce. Úkolem bylo vytvořit docházkový systém, který na základě vyplnění jména a příjmení spolu s činností, která bude v laboratoři vykonávána, umožní spuštění laboratoře. Samotné měření je úkolem jiného diplomanta, který vše vytváří v programu LabView. Zároveň se všechny informace, které budou vyplněny, uloží do testového souboru a ten se pak bude pomocí FTP stahovat do počítače. Tímto by se měl usnadnit systém vedení docházky, který je zatím veden papírově.

Další úkol vzešel z mého popudu, kdy jsme se bavili o značení měřených fotovoltaických panelů z vedlejší laboratoře. Z prohlášení, že vytváření štítků trvá na celém procesu nejdéle, jsem nabídl možnost přidání termotransferové tiskárny do možností PLC. Škola zakoupila tiskárnu Zebra GK420T, která bude připojena k PLC a přes ovládací panel se bude možnost tisknout štítky podle potřeby.

Nejdříve tedy popíšu tvorbu programu, který bude cyklicky zpracováván v PLC a pak prostředí pro ovládací panel. Popsána bude i komunikace mezi PLC a tiskárnou, stejně jako příkazy, které bude tiskárna zpracovávat.

### 4.1.1 Hardwarová konfigurace

Prvním krokem k tvorbě programu v softwaru Simatic STEP 7 je vytvoření projektu a následně přidání hardwarové konfigurace, která se později nahraje do CPU, v mém případě CPU 315-DP. V hardwarové konfiguraci je skryté adresování jednotlivých karet a nastavení procesoru. V praxi to funguje jednoduše, neboť je software uživatelsky přívětivý s funkcí drag and drop. Všechny modulové karty jsou k nalezení v knihovně a vyhledávání funguje pomocí sériového označení.



Obr. 22 HW konfigurace

Ve spodní tabulce je znázorněno adresování vstupů a výstupů na jednotlivých DI/DO kartách. V mém případě jsem pro první kartu zvolil nulový a první byte pro vstupy i výstupy. To ve finále znamená, že první vstup na první kartě bude mít adresu I0.0 a první výstup analogicky Q0.0. Další DI/DO karta zabírá čtvrtý a pátý byte, a to z prostého důvodu, měl jsem již vytisknuté popisky na karty a nenašel jsem druhý a třetí byte.

Další nastavení proběhlo u karty pro sériovou komunikaci CP340, kdy je pro příjem a odesílání dat zapotřebí šesti bytů vstupů a výstupů. Adresy byly automaticky vygenerovány programem a neměl jsem potřebu je měnit.

Nastavení CPU už poskytuje více možností, nejprve se zaměřím na nastavení retentive memory. CPU 315-2 DP má k dispozici 2048 bytů a všechny se dají nastavit jako retentive. Já budu ukládat všechny data do datových bloků, takže jich nepotřebuji tolik, nechám tedy automaticky nastavených 16 bytů. Dále můžu nastavit počet nastavitelných timerů a counterů, u obou mi jich bude stačit deset. Z datasheetu lze dále vyčíst, že každý block má maximální velikost lokální paměti až 2048 bytů.

Data areas and their retentivity	
retentive data area in total	All, 128 KB max.
Flag	
• Number, max.	2 048 byte
• Retentivity available	Yes; MB 0 to MB 2 047
• Retentivity preset	MB 0 to MB 15
• Number of clock memories	8; 1 memory byte
Data blocks	
• Retentivity adjustable	Yes; via non-retain property on DB
• Retentivity preset	Yes
Local data	
• per priority class, max.	32 768 byte; Max. 2048 bytes per block

Obr. 23 Část datasheetu CPU

Dále je potřeba nastavit nějaký byte z M-paměti jako memory clock byte. Tím si do určitého bytu nastavím, že každý bit bude periodicky měnit svou binární hodnotu, akorát s jinou frekvencí. Funkce těchto memory clock bytů se dá využít pro signalizaci.

V tabulce níže je uvedeno, který bit bude měnit hodnotu s jakou frekvencí.

Bit of the Clock Memory Byte	7	6	5	4	3	2	1	0
Period Duration (s)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Frequency (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

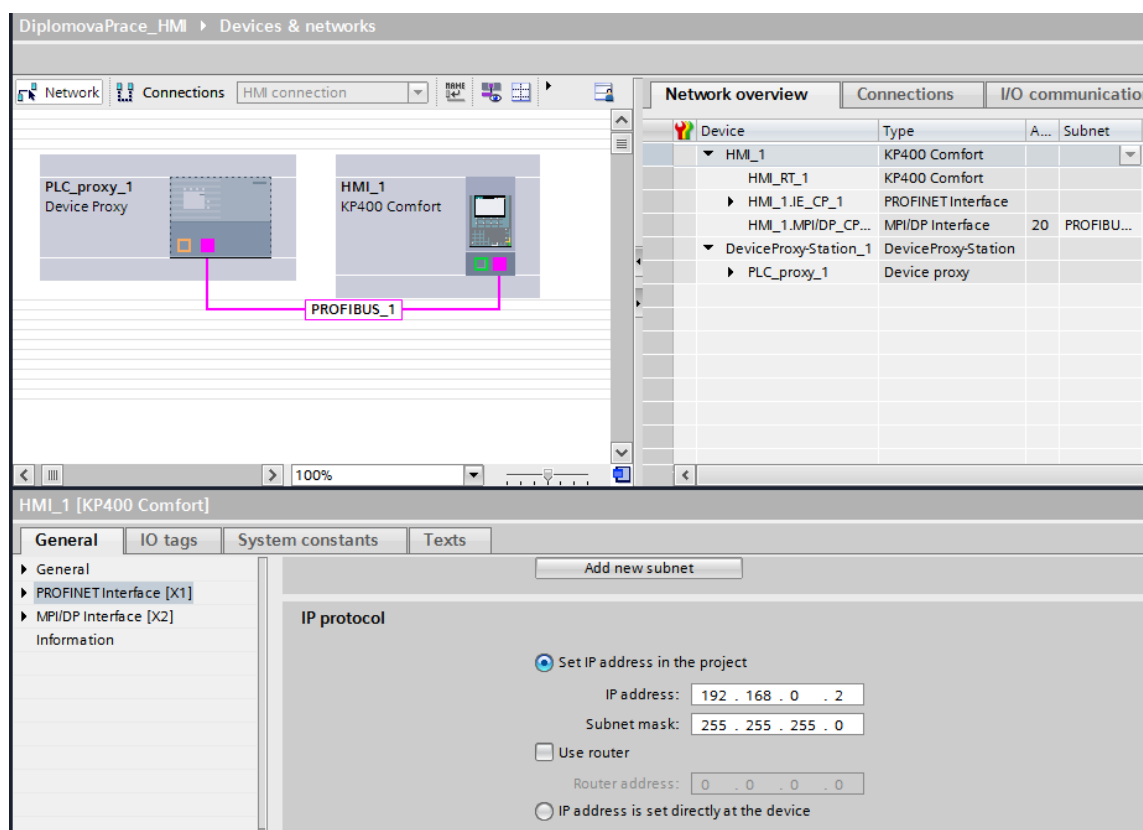
Tab. 6 Memory clock byte

Déle proběhne nastavení komunikace PLC s ovládacím panelem a programovací jednotkou, tedy mým počítačem. Toto CPU má k dispozici dva porty, jeden MPI (Multi Point Interface) a druhý ProfiBus. ProfiBus budu využívat pouze na komunikaci s panelem, nastavené hodnoty již byly popsány. MPI bude sloužit pouze k nahrávání programu do stanice. [12]

## 4.1.2 Konfigurace operačního panelu

Díky tomu, že se projekt vyvíjí zároveň ve dvou různých softwarech, je nutné vytvořit hardwarovou konfiguraci i pro operační panel. Zatímco ve STEP 7 se vytvářela HW konfigurace pouze pro PLC, v TIA Portal se musí vytvořit konfigurace panelu a konfigurace síťového propojení. Jelikož jsem nastavení PLC již vytvořil ve druhém programu, nemusím ho nastavovat znovu. V TIA Portal si vytvořím Proxy PLC a stáhnu si konfiguraci z již vytvořeného projektu.

Zároveň si přidám operační panel, znovu podle sériového označení. Panel KP400 má port, který lze nastavit jako MPI nebo jako port ProfiBusu. Tento port bude využíván ke komunikaci s PLC, takže bude nastaven jako ProfiBus DP s adresou 20. Panel má zároveň port pro ProfiNet, který budu využívat primárně k nahrávání programu a po dokončení projektu ke stahování sběru dat. Adresa byla nastavena na statickou 192.168.1.3, později bude možné využívat routeru k dynamickému přidělování adresy. Síťové propojení je znázorněno na obrázku č. 24, které je také možné vidět přímo v projektu TIA Portal.



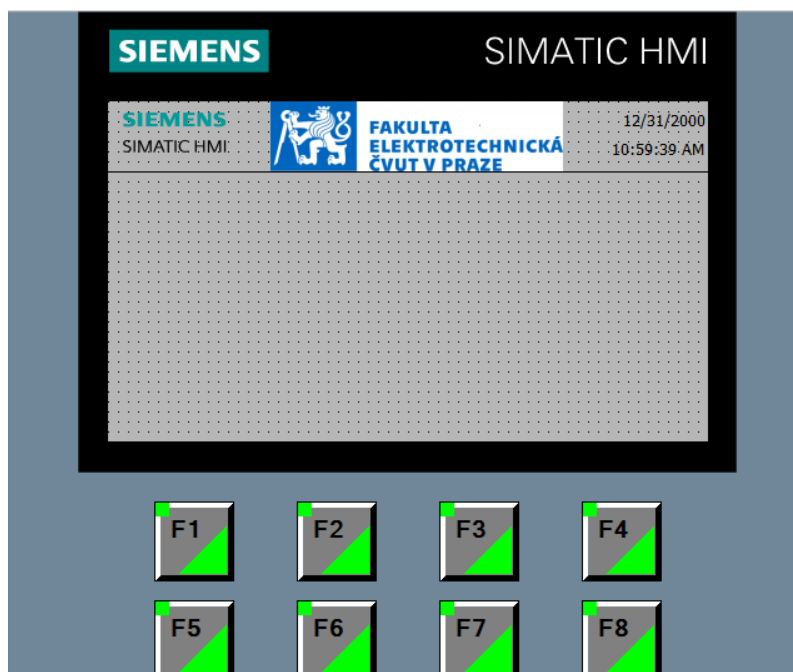
Obr. 24 Síťové propojení v TIA Portal

## 4.2 Funkce automatizace

Hlavní funkcí automatizace je spuštění laboratoře po přihlášení do systému. Původním záměrem bylo přihlašování pomocí hesla, nicméně od toho záměru jsem upustil z několika důvodů. Panel má určitý limit špatných přihlášení do systému a zároveň klávesnice, ze které se text zadává, není z nejkomfortnějších, což by akorát vedlo k častému zablokování. Odblokovat systém lze pouze pomocí počítače s licencovaným programem, který škola jednoduše nemá k dispozici.

Přihlášení pomocí RFID čipů jsem však nezavrhl v tomto projektu úplně. Počítá se s tím, že čtečku vytvoří jiný student a proto do PLC přidám funkci, která bude obsluhovat komunikaci, ukládání uživatelů a přihlašování. Tato funkce bude zatím neaktivní, v menu pak půjde přepínat mezi obyčejným přihlášením a přihlášením pomocí zaměstnanecké karty.

Tím pádem jsem se rozhodl, že pro spuštění bude stačit výběr uživatele, kterých může být až 10 a přidávat je bude moci osoba oprávněná k tomuto úkonu. Přidávání uživatelů a ostatní nastavení bude možno až po autorizaci klíčem, který bude mít domluvená osoba. Mezi další nastavení bude patřit konfigurace tiskárny, změna data a času v PLC a volba napájení laboratoře.



Obr. 25 Šablona obrazovky HMI



Všechny obrazovky budou vytvořeny podle jednotné šablony, na níž je zobrazeno logo fakulty spolu s logem Siemensu a datem a časem. Pod obrazovkou jsou tlačítka, kterým byla přiřazena určitá bitová adresa, která bude stiskem měnit hodnotu. Toho se dá později využít v programu PLC. Šablona je zobrazena na obrázku č. 25.

#### 4.2.1 Spuštění laboratoře

Po příchodu do laboratoře bude uživatel přes ovládací panel vyzván ke stažení tlačítka START, které bude umístěno u vchodu. Po stlačení tlačítka bude uživatel přesměrován na přihlašovací stránku, kde se bude moci autorizovat pomocí rolovacího seznamu a následně vyplnit popis vykonávané činnosti v laboratoři.



Obr. 26 Obrazovka přihlášení

Na následujícím obrázku č. 27 je úryvek z kódu z funkčního bloku FB1 – Zapnutí laboratoře, který se stará o výše zmíněnou funkci. Nejprve porovnávám, že jsem na startovací obrazovce pomocí wordu, do kterého HMI ukládá aktuální obrazovku. Konkrétní word se dá nastavit v HW konfiguraci panelu.

Dále se cyklicky kontroluje, jestli není sepnutý Emergency STOP a čeká se jen na stisknutí tlačítka START. Náběžná hrana je zde kvůli možné poruše tlačítka, jako je například slepení kontaktu, což je častý neduh těchto spínačů.

```

Network 1 : Spuštění tlačítkem START
A(
L   "Komunikace s panelem".OP_GetScreenNumber   DB10.DBW14
L   100
=I
)
A(
A   "I   0.0"                                     IO.0           -- Tlačítko START
FP  #Hrany[1]                                    #Hrany[1]
)
AN  #Spusteno                                    #Spusteno
A   #EM_STOP                                    #EM_STOP
=   L   0.0

A   L   0.0
R   #Vypnuto                                    #Vypnuto
A   L   0.0
S   #Spusteno                                    #Spusteno

A(
ON  #EM_STOP                                    #EM_STOP
O   #Vypnuto                                    #Vypnuto
)
R   #Spusteno                                    #Spusteno
A   #Spusteno                                    #Spusteno
=   L   0.1
A   L   0.1

L   101
T   "Komunikace s panelem".OP_SetScreenNumber   DB10.DBW12

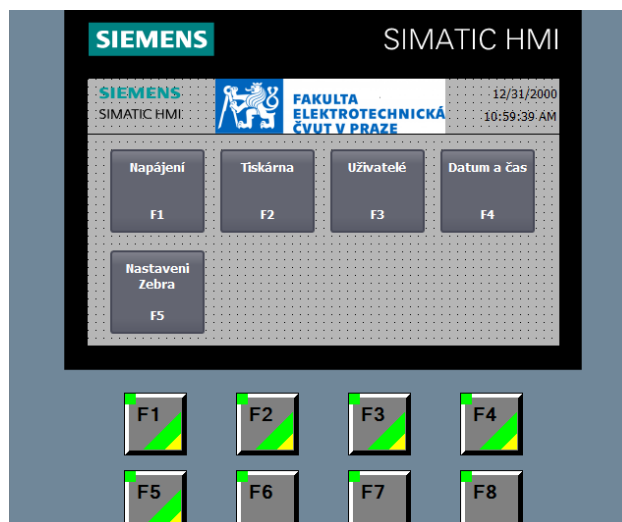
A   L   0.1
FP  #Hrany[5]                                    #Hrany[5]
L   0
T   "Komunikace s panelem".Uzivatel             DB10.DBW20

A   L   0.1
S   "Q   0.0"                                    Q0.0           -- START - LED

```

Obr. 27 Úryvek kódu z FB1

Po stisknutí si změním hodnotu mezivýsledku #Spusteno na logickou jedničku a zároveň resetuji hodnotu #Vypnuto, která zůstává v logické jedničce po posledním vypnutí. Po spuštění se automaticky změní obrazovka pomocí nastaveného wordu „OP\_SetScreenNumber“, také se defaultně nastaví 0 do proměnné uživatel, tedy Nevybráno, jako je na obrázku č. 26. Poslední činností prvního networku FB1 je spuštění napájení LED, které vytvoří podsvícení tlačítka START.



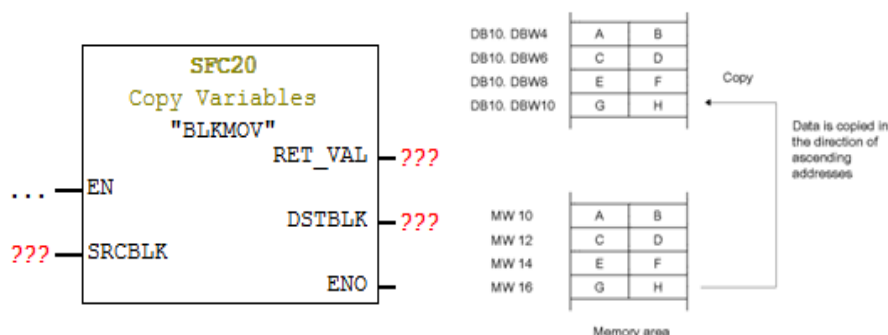
Obr. 28 Menu

Po spuštění bude možné přejít i do menu, ze kterého se bude moci uživatel dostat do nastavení napájení, tiskárny štítků a jejího nastavení, přidávání uživatelů a změny data a času. Na všechny screeny se lze dostat pomocí tzv. hotkeys F1 až F5. Screen Uživatelé bude zpřístupněn až po autorizaci klíčem, zbylé obrazovky budou zpřístupněné ihned po zapnutí laboratoře. Menu je zobrazeno na obrázku č.

## 4.2.2 Přihlášení do laboratoře

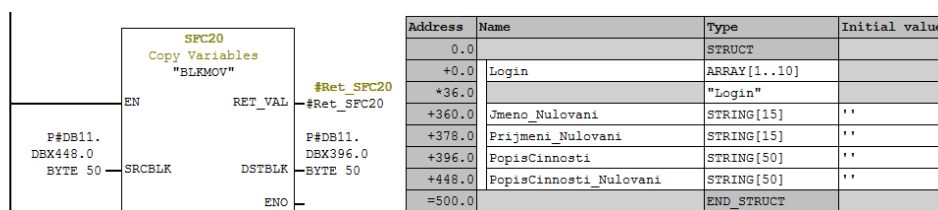
Pro zapnutí laboratoře je nutné vybrat uživatele z rolovacího seznamu na přihlašovací obrazovce (obrázek č. 26 vpravo) a vyplnit krátký popis činnosti. Bez přihlášení nebude možné laboratoř zapnout a tlačítko „Zapnout laboratoř“ bude zašedlé. Po výběru ze seznamu uživatelů a vyplnění popisu činnosti se tlačítko „Zapnout laboratoř“ rozbliká s frekvencí 0,5 Hz, což bude signalizace správného vyplnění polí a povolení zapnutí laboratoře. Po zapnutí se políčko rozsvítí zeleně a spustí se napájení laboratoře.

Uživatelé jsou uloženi v databázi DB11 LoginDB, přidávat a měnit je bude možné ze screenu Uživatelé, z HMI si PLC bude vyčítat pouze číslo uživatele, který bude vybrán. Popis činnosti se bude ukládat do stejné databáze, zároveň je v databázi umístěno nulování jména, příjmení a popisu činnosti. Důvod je prostý, Simatic nemá funkci vyložení na inicializování databází (všechny hodnoty by se tak přepsaly startovací hodnotou) a i kdyby měl, tak bych inicializoval všechny proměnné včetně uložených uživatelů a při každém startu by se musely ukládat znovu, což není zrovna nejšťastnější řešení. Proto jsem využil systémové funkce SFC20 neboli BLKMOV (Block Move), která funguje ke kopírování dat z jedné oblasti paměti do druhé.



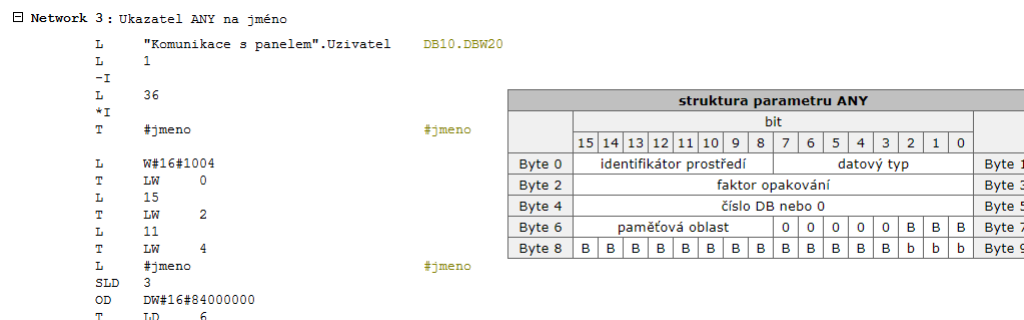
Obr. 29 Systémová funkce BLKMOV

SFC20 lze adresovat pomocí parametrizovaného a neparametrizovaného ANY pointeru, já budu využívat obou možností. K nulování využívám neparametrizovaného ANY pointeru, kdy např. na vstup SRCBLK pro nulování popisu činnosti přijde pointer na první bit nulovacího stringu z DB11 spolu s délkou stringu a na výstup DSTBLK obdobně první bit nulovaného stringu, také se jeho délkou. RetVal pouze zaznamenává chyby, které při kopírování můžou nastat.



Obr. 30 Použití funkce SFC20

Parametrizovaného ANY pointeru využívám při ukládání dat, které pak použiji ke sběru dat pro docházkový systém. Parametr ANY se používá tehdy, když je datový typ aktuálního parametru neznámý nebo pokud je možno použít libovolný typ dat. ANY, jako libovolný parametr typu ukazuje na začátek datové oblasti a určuje délku oblasti. Parametr ANY zabírá délku 10 Byte paměti a má jasně danou strukturu, které lze využít k parametrizaci, což umožní dynamické změny oblasti paměti. Na obrázku níže je zobrazena struktura spolu s mým použitím.



Obr. 31 Ukazatel ANY a jeho struktura

Nejprve si vyčtu z databáze DB10, který uživatel byl zvolen v podobě integeru v rozsahu od jedné do desíti. Také víme, že ačkoliv má string určitý počet bytů, je rozšířen o tzv. hlavičku, která sama obsahuje dva byty. Skutečnou a maximální délku řetězce, v mém případě jméno zabírá 18 bytů včetně hlavičky a po jménu následuje hned příjmení. Tím pádem mi každé jméno začíná na bytech s násobkem 36 včetně nuly, a pomocí tohoto výpočtu si při každém přihlášení vytvořím offset pro ukazatel na jméno uživatele.

Poté začnu sestavovat ANY pointer, který bude pouze lokální proměnnou funkce FC 1 Logovani, podle definované struktury. Do nultého localwordu si uložím W#16#1004, což jsou v podstatě dva byty, první s hodnotou B#16#04 a druhý B#16#10. První je identifikátor prostředí (10 pro S7) a druhý značí datový typ byte. Do dalšího localwordu uložím číslo 15, což je v tabulce nešikovně označeno jako faktor opakování, přičemž lepší popis je počet elementů datového typu v oblasti. Další možností je výběr datového bloku (u mě DB11) nebo odkazování na jinou oblast, když zde uloží nula. Pokud se zde uloží nula, je nutné odkazovat, aby se změnila paměťová oblast podle tabulky níže.

Hexadecimal Code	Coding of the Memory Areas	
	Area	Description
b#16#80	P	I/O area
b#16#81	I	Input area
b#16#82	Q	Output area
b#16#83	M	Bit memory area
b#16#84	DB	Data block
b#16#85	DI	Instance data block
b#16#86	L	Local data (L stack)
b#16#87	V	Previous local data

Tab. 7 Paměťové oblasti

Já využívám kopírování z datového bloku, tudíž se mě týká B#16#84 jako kód paměťové oblasti. Toho využiji spolu s offsetem, který ještě posunu v doublewordu o tři místa doleva. Posunutý offset pak maskuji funkcí OW (OR Word) s DW#16#84000000 (paměťová oblast DB) a výsledek uložím do posledního DW ANY pointeru. Tímto získám dynamický offset pro kopírování dat z oblasti, kterou definuje vybraný uživatel.

### 4.2.3 Docházkový systém

Výše uvedených funkcí využiji při ukládání hodnot a logování. Při zapnutí laboratoře se spustí požadavek na sběr dat. Díky požadavku si pomocí SFC20 a parametrizovaného ANY pointeru překopíruji jméno a příjmení aktuálního uživatele spolu s popisem činnosti do databáze DB15 OP\_Logovani.

Po překopírování nastavím request pro ukládání do logické jedničky, viz obr. x. Hodnota requestu se cyklicky zpracovává ve skriptu, který běží na pozadí programu HMI. Při změně hodnoty requestu se pak spustí skript Logovani, který pak zapisuje do textového souboru na SD kartě ovládacího panelu. Tím, že se skript spouští změnou hodnoty, lze pro zapnutí a vypnutí laboratoře využít jednoho bitu. Skript bude při změně hodnoty požadavku na logickou 1 ukládat čas zapnutí, jméno a příjmení uživatele a popis činnosti a při změně hodnoty na logickou 0 čas vypnutí a text „Ukončení činnosti“.

Skript funguje jednoduše, při zapnutí a změně bitu „Ulozit“ se ptám jestli se log zapíná. Pokud ano, vytvořím si cestu k souboru /Storage Card SD/DOCHAZKA.TXT, jestli soubor neexistuje, tak se na krátko vytvoří nový se stejným názvem. Poté se sestavuje string z aktuálního času v PLC, uživatele a činnosti v laboratoři, vše je odděleno středníkem. Po sestavení a uložení řetězce se soubor zavře a nastaví se bit „Uloženo“, kterým pak umožňuji v programu PLC další činnosti.

```

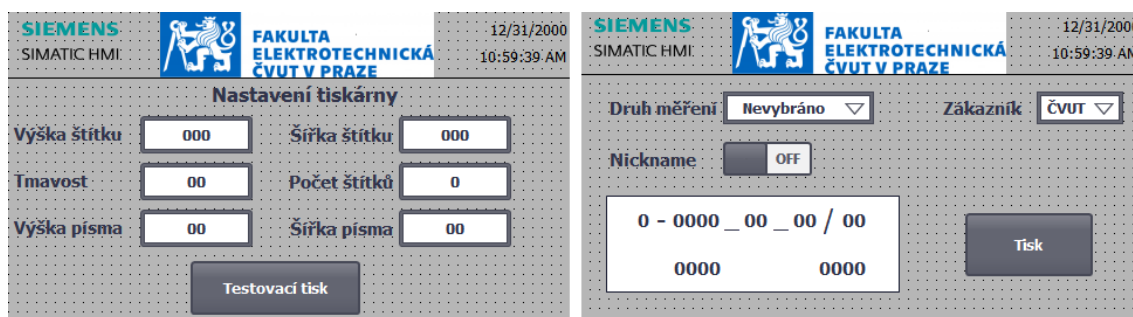
1 Sub Logovani()
2
3 Dim file
4 Dim path
5 Dim mode
6 Dim delimiter
7 Dim gap
8
9 If SmartTags("OP_Logovani_Logovani_Ulozeno") = True Then
10 Exit Sub
11 End If
12
13 If SmartTags("OP_Logovani_Zapnout_Log") = True Then
14
15 path = "\Storage Card SD\DOCHAZKA.TXT"
16 SmartTags("Nazev_souboru") = "DOCHAZKA.TXT"
17 SmartTags("Path_Nazev_souboru") = path
18
19 delimiter=";"
20 gap=" "
21
22 'Starting error routine
23 On Error Resume Next
24 'Create a file object
25 Set file= CreateObject("FileCtl.File")
26
27 'Check if any errors happend
28 If Err.Number<> 0 Then
29 ShowSystemAlarm "Error # " & CStr(Err.Number) & " " & Err.Description
30 Err.Clear
31 Exit Sub
32 End If
33
34 'Open or create a file
35 file.open path,mode
36 'Check if any errors happend
37 If Err.Number<> 0 Then
38 ShowSystemAlarm "Error # " & CStr(Err.Number) & " " & Err.Description
39 Err.Clear
40 Exit Sub
41 End If
42
43 'Write data into file
44 file.LinePrint(Now & delimiter & SmartTags("OP_Logovani_Uzivatel_Jmeno") & gap & SmartTags("OP_Logovani_Uzivatel_Prijmeni"
45 & delimiter & SmartTags("OP_Logovani_PopisCinnosti"))
46 'Close the file
47 file.Close
48 'Release the Object
49 Set file = Nothing
50 SmartTags("OP_Logovani_Logovani_Ulozeno") = True
51 End If

```

Obr. 32 Funkce Logování

## 4.2.4 Funkce tiskárny

Další funkcí PLC je tisk štítků na termotransferové tiskárně. Na štítku by se mělo objevit označení měření (mezera pro fotovoltaiku, B – baterie a U – UCEEB). Dále datum vystavení zakázky, označení měření (01 až 89 pro cizí zakázku, 90 až 99 pro ČVUT). Na dalším řádku může být libovolně vlastní označení (nickname) a rozlišení. Vše bude možné nastavit na obrazovce tiskárny viz. obrázek níže. Další možností je změna parametry tisku, jako je velikost písma, rozměry štítku, tmavost potisk a počet štítků.



Obr. 33 Obrazovka tiskárny a jejího nastavení

Komunikaci obstarává karta Siemens CP340, která přes RS232 posílá mnou složený kód. Ten je psán v jazyce ZPL, což je soubor příkazů přímo od společnosti Zebra vytvořených právě pro ovládání jejich tiskáren. Posílaný kód je znázorněn a popsán na obrázku níže.

```

^XA //Start
^MMT //Režim Tear-off
^ALL150 //Výška štítku
^PW650 //Šířka štítku
^MD20 //Tmavost
^LS4 //Posun
^F030,30 //Souřadnice prvního řádku
^ADN35,25 //Velikost písma prvního řádku
^FDB-2019_05_08/01^FS //Text prvního řádku
^F030,80 //Souřadnice druhého řádku
^ADN35,25 //Velikost písma druhého řádku
^FD0001 €€€€^FS //Text druhého řádku
^PQ1,0,1,Y //Počet štítků
^XZ //Konec

```

Obr. 34 Kód ZPL pro tiskárnu

Celý řetězec je umístěna v databázi DB13 - DataTiskárna a je sestavován ve funkci FC2 – SkladaniStringu. Většinu proměnných lze nastavit z obrazovky „Nastavení tiskárny“. Při spuštění obrazovky „Tiskárna“ je datum předvoleno na datum aktuální a zbylé položky podle posledního známého stavu.

## 4.2.5 Vedlejší funkce

Mezi vedlejší funkce lze zařadit ovládání napájení. Může se stát, že bude potřeba mít některé z napájecích zásuvek nebo stanic aktivní i při vypnuté laboratoři. Proto lze na screenu „*Napájení*“ nastavit, která zásuvka má být pod napájením i při vypnutém stavu. Tato funkce je jednoduchá a není potřeba ji popisovat, nicméně neuškodí ji mít v záloze. Zásuvky se budou spínat podle posledního známého stavu tak, aby uživatel tuto funkci využíval co nejméně.

Funkcí, která se v budoucnu bude určitě hodit je nastavení data a času. Simatic nepočítá se změnou času a také se stává, že se někdy skutečný čas a čas v PLC rozchází, což je při sledování docházky atribut, který se moc nehodí. Datum a čas lze nastavit na obrazovce „*Datum a čas*“ (obr. výše) a o funkci se stará funkční blok FB2 DateTime, ve které se volají další systémové funkce. První systémovou funkcí volanou v FB2 je SFC1 (Read System Clock), která čte čas PLC a ukládá ho do proměnné formátu Date-and-Time o délce 8 bytů.

```

□ Network 5 : Vyčítání data a času
LAR1 P##Set_DT
L   #Year_HMI          #Year_HMI
ITB
T   B [AR1,P#0.0]

L   #Month_HMI        #Month_HMI
ITB
T   B [AR1,P#1.0]

L   #Day_HMI          #Day_HMI
ITB
T   B [AR1,P#2.0]

L   #Hour_HMI         #Hour_HMI
ITB
T   B [AR1,P#3.0]

```

Obr. 35 Vyčítání data a času pomocí SFC1

Z této proměnné si pak cyklicky po bytech vyčítám rok, měsíc, den, hodinu a minutu času v PLC. Tyto proměnné využívám ve funkci tiskárny. Zároveň se na stejné obrazovce může čas a datum nastavovat, a to za pomoci SFC0 (Set System Clock). Jelikož se na nastavovací obrazovce čísla zapisují v jiném formátu, musí se ještě využít převodu každého čísla z integeru na byte a poté zapsat do proměnné #Set\_DT. Po zmáčknutí tlačítka „Nastavit“ si SFC0 stáhne hodnotu poskládané proměnné #Set\_DT a nastaví si podle ní datum a čas.



Nicméně tato funkce je jen dočasným řešením, v budoucnu se plánuje vytvoření NTP serveru, který bude spuštěn na hlavním počítači. Podle něj se budou synchronizovat ostatní členy laboratoře, včetně PLC.

### 4.3 Instalace

Po delším testování funkčnosti a simulování různých stavů lze přejít k instalaci automatizace do rozvaděče a následné implementaci do laboratoře. Nicméně tato část realizace je časově náročná a na chvíli vyřadí laboratoř z provozu. Jde o sestavení rozvaděče, připojení do sítě a kompletní přepojení silové části laboratoře. Momentálně se zdá, že je nereálné vše stihnout do odevzdání písemné části diplomové práce, tudíž zde není a nebude zdokumentován proces instalace.

Nicméně se budu snažit vše dokončit v intervalu mezi odevzdáním práce a státní závěrečnou zkouškou, tak aby byla k obhajobě diplomové práce připravena dokumentace i praktická část. Pokud bude možné dokončit instalaci před obhajobou, bude instalace dokončena v nejbližším možném termínu.

## 5 Bezpečnostní analýza

V rámci bezpečnostní analýzy je nutné nahlížet na automatizaci ze dvou pohledů. Zaprvé se bezpečností myslí ochrana personálu obsluhujícího laboratoř a zadruhé zabezpečení dat a autorizace práce v laboratoři.

### 5.1 Bezpečnost laboratoře

Jelikož se jedná o laboratoř pro měření baterií, hlavním rizikem jsou úrazy elektrickým proudem a popálení. Činnost v laboratoři je tak řízena požadavky pro bezpečnou práci v laboratořích podle normy ČSN 01 8003. Elektrická zařízení musí být udržována ve stavu, který odpovídá platným elektrotechnickým předpisům a normám. Elektrická zařízení se musí revidovat a přezkušovat v rozsahu a lhůtách stanovených příslušnými normami a směrnici výrobce. Obsluha a práce na elektrických zařízeních se řídí ustanoveními uvedenými v ČSN 33 1310 ed. 2, ČSN EN 50110-1 ed. 3 a ČSN EN 60745-1 ed. 3.

Ochrana při měření baterií je však již implementována za pomoci krytu, který je umístěn kolem baterie při měření. K této ochraně je u každého stanoviště umístěno vlastní STOP tlačítko, které při stlačení odepíná napájení stanoviště, nicméně tento systém není ničím řízen a neodepíná napájení zásuvky. Tento problém je řešen v automatizaci, kdy jsou všechny zásuvky spínány stykači. Nouzové tlačítko bude umístěno na snadno dostupném místě u dveří a při stlačení odepne napájení celé laboratoře.

Odepnutí probíhá tak, že se přestanou napájet výstupní karty PLC, které řídí jednotlivé stykače pomocí 24V relé. Každé použití nouzového tlačítka je snímáno na vstupní kartě PLC a signál o něm vyvolá zápis do logu. Obsluha laboratoře bude mít tak záznam času, kdy porucha nastala a po jaké době se problém vyřešil a laboratoř se zase spustila. Při analýze rizik tohoto řešení jsem došel k závěru, že jedinou hrozbou by mohla být porucha elektrických přístrojů (slepení kontaktů relé nebo stykače atd.). Toto riziko je však minimalizováno při použití polovodičového SSR relé, které je sice dražší, ale je zbaveno rizik obyčejného relé.

Toto odepnutí však přichází až po zásahu obsluhy za předpokladu, že nezasáhla jedna ze základních ochran. Tím je myšlena soustava jističů a chrániče, která je nadřazena všem přístrojům v laboratoři.

## 5.2 Zabezpečení laboratoře

Původním záměrem byla autorizace pomocí RFID čipů v zaměstnaneckých kartách, nicméně při tomto řešení jsem byl limitován dostupnými čtečkami, který by splňovaly nutné náležitosti. Tato možnost je však stále splnitelná a v PLC jí nic nebrání, pokud se taková čtečka vytvoří nebo koupí. Spolu s pořízením čtečky je nutné zakoupit další kartu CP340 pro sériovou komunikaci.

Další možností po RFID čipech byla autorizace uživatelským jménem a heslem, ta by pak umožnila spuštění a práci v laboratoři. Od tohoto konceptu jsem však upustil z prostého důvodu. Operační panel má svou alfanumerickou klávesnici, která přímo vybízí k zadání špatného hesla, což v kombinaci s pevně daným počtem špatných přihlášení automatizaci snadno zablokuje. Odblokování vyžaduje software, který škola nemá k dispozici. Další nevýhodou je přidávání nových uživatelů, ke kterému je potřeba stejného softwaru jako v předchozím případě.

Proto jsem se rozhodl pro autorizaci ze seznamu vedoucím přidáných uživatelů. U těchto uživatelů se předpokládá, že mají svůj klíč od laboratoře, což už je samo o sobě zabezpečení laboratoře. Uživatel, který aktuálně v laboratoři pracuje, je spolu s činností ukládán do logu. Přístup do logu je však už podmíněn heslem, které bude znát pouze vedoucí pracovník.

## 5.3 Připojení do sítě

Ethernetový port je přítomný pouze u operačního panelu KP400 Comfort, zatímco CPU má k dispozici ProfiBus a MPI. To ve výsledku znamená, že bude do sítě připojen pouze panel, což aplikaci vyhovuje z toho hlediska, že je z PLC potřeba pouze číst log docházkového systému. Zatím není třeba připojovat PLC na vzdálenou správu. Nicméně tato možnost existuje pomocí vlastního routeru od Siemensu řady SCALANCE a programu Sinema RC Client. Ten v podstatě na dálku vytvoří VPN tunel mezi počítačem a PLC. Této možnosti by se dalo využít, nicméně na pouhé ovládání laboratoře mi přijde zbytečně nákladná.

Operační panel bude připojen do lokální sítě laboratoře s IP adresou 192.168.1.3 a maskou 255.255.255.0. Komunikace v lokální síti je směrována pomocí routeru a je zprostředkována pouze mezi členy se síťovým rozhraním bez přístupu na Internet. Ten má pouze hlavní počítače, který je pak přístupný přes TeamViewer.

### 5.3.1 Připojení z počítače

K připojení počítače do PLC je vyžadováno vícero věcí. Zaprvé licencovaný program Siemens STEP 7 spolu s aktuálním programem, který je právě v PLC. Bez něj se totiž může uživatel připojit, ale jedinou možností, jak v PLC něco měnit je upload programu z PLC. V předchozích kapitolách bylo rozebráno, že PLC má vyhrazenou paměť na ukládání programu, nicméně neukládá popisky, komentáře ani symboliku. Upload tedy pak vypadá jako holý text s absolutním adresováním, tudíž uživatel nepozná, o která data se jedná a jakákoliv změna je pak spíše kontraproduktivní.

Další potřebnou součástí nahrávání a modifikace programu je převodník Ethernet/MPI, tedy pokud počítač nemá vlastní MPI port. Tento převodník není z nejlevnějších, takže vyvstává otázka, jestli se připojení z hlavního počítače vyplatí. Osobně bych byl pouze pro to, aby se uživatel připojoval pouze za účelem změny programu. Dokud není PLC propojeno s LabView, není třeba číst z PLC jiná data než log docházky. Stálé připojení PLC k počítači by bylo vhodné, pouze pokud by se měnil program nějak často.

Stálé připojení bude mít jen HMI, přes které se do PLC uživatel nedostane. Toto spojení bude k dispozici pouze pro stahování přes webové rozhraní, které panel nabízí. Jak již bylo řečeno, nová řada panelů od Siemensu nabízí tvorby HTML serveru, ze kterého lze stahovat uložená data. Webové rozhraní je znázorněno na obrázku č. 36. Uživatelské jméno pro přístup do složek je „log“ a heslo si definuje uživatel.

The screenshot displays the SIMATIC HMI Miniweb interface. At the top, the Siemens logo is visible on the left, and the title "SIMATIC HMI Miniweb on HMI\_Panel" is centered. Below the title, there is a "Miniweb Start Page" section. On the left side, there is a navigation menu with the following items: "Start page", "Remote Control", "Control Functions", "System Diagnostics", and "File Browser". The "Start page" is currently selected. The main content area shows a "Welcome on HMI\_Panel" message and a "Device Status of HMI\_Panel" section indicating "The runtime is running". Below this, there is a table titled "General Device Information" with the following data:

General Device Information	
Device Type	TP700 Comfort
Image version	V13.00.01.09_03.01
Bootloader version	1.16
Bootloader release date	2.9.2013
Device Name	HMI_Panel
S\$LEAY_VERSION	OpenSSL 1.0.1j 15 Oct 2014

At the bottom of the page, there is a "Hint" section: "Hint: When the devicename contains an underscore ( \_ ) some browsers have a bug that makes it impossible to log in. One possible solution may be to use the IP address of the device instead of the name, or to use another browser."

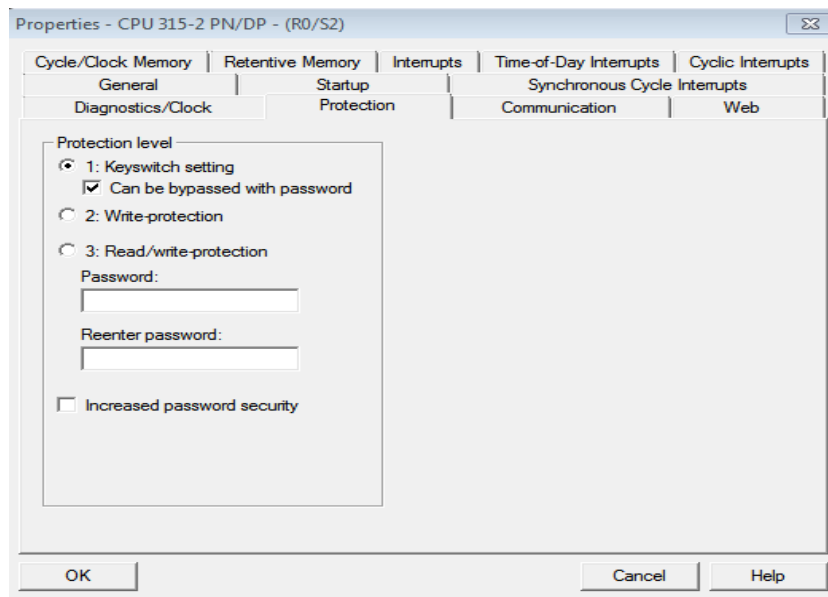
Obr. 36 Webové rozhraní HMI

## 5.4 Kyberbezpečnost

V první řadě je nutné analyzovat rizika, která mohou nastat při nabourání do systému. Jelikož se jedná o jednoduché řízení laboratoře, rizika jsou následující:

- Smazání programu PLC s následkem nefunkční laboratoře
- Po propojení s LabView stažení a krádež naměřených dat
- Spuštění bez autorizace simulováním vstupů
- Stažení osobních údajů uživatelů (Jméno, RFID čip zaměstnanecké karty)

Všechna tato rizika jsou podmíněny připojením počítače do PLC a následnou modifikací programu (simulováním vstupů, mazáním a kopírováním databází atd.). Tomuto lze zabránit jediným způsobem a to spuštěním ochrany PLC v hardwarové konfiguraci, viz obr. 37. Spuštěním ochrany se rozumí požadavek autorizace při jakémkoliv připojení počítače do PLC. Tím pádem se v rámci programu PLC zamezí všem rizikům z analýzy jedním tahem. Toto elegantní řešení je jednou z výhod a důvodů, proč jsem vybral právě Simatic.



Obr. 37 Ochrana PLC

Ochrana programu HMI není nutná, jediná data se totiž ukládají v paměti panelu a jediná možnost jak je získat je přes HTML Server, který je sám o sobě zaheslovaný. Existuje sice ještě jedna možnost a to stažení dat přes USB port, nicméně ztráta docházkového systému není žádné enormní riziko.

## 5.5 Rozdíl oproti stávajícímu řešení

Stávající řešení nemělo žádné řízení, docházkový systém byl veden papírově a bezpečnost individuálně pro každou stanici. Má práce tedy usnadňuje činnost v laboratoři, zlepšuje bezpečnost a zabezpečení. Všechna spuštění, nouzová zastavení a ukončení budou logována do přehledného textového souboru uloženého na SD kartě operačního panelu. Přidávání uživatelů je řešeno jednoduše přes HMI pomocí autorizace klíčovým přepínačem, který u sebe bude mít pouze vedoucí.

## 6 Zhodnocení práce

Cílem mé práce bylo analyzovat potřeby zkušební laboratoře z hlediska automatizace jejího provozu. Navrhnout vhodnou strukturu (HW a SW) pro řízení laboratoře s využitím výstupů analýzy a s ohledem na již používaná ve spolupráci s katedrou. V první části jsem hlavně sbíral informace a potřeby katedry, poté jsem se přesunul k návrhu. Po otestování a před instalací ještě došlo k analýze rizik a zabezpečení laboratoře.

### 6.1 Možná vylepšení

Má diplomová práce nabízí celou řadu možných rozšíření. Prvním je realizace přihlášení pomocí RFID čipů, což bylo původním záměrem. Toto rozšíření by usnadnilo přihlašování a rozhodně by zlepšilo zabezpečení laboratoře proti neautorizovaným zásahům. Pro toto rozšíření je již připraven základ pomocí funkce, která bude komunikaci s RFID čtečkou obsluhovat.

Další možností je propojení se systémem LabView, který se stará o měření. PLC by si z něj mohlo odečítat naměřené hodnoty a ukládat je spolu s docházkovým systémem. Také stojí za uvážení přidání čidel na monitoring podmínek v laboratoři (teplota, vlhkost atd.), které by se ukládaly při spuštění. Simatic dále nabízí spoustu dalších možností, např. přidání GSM modulu pro zasílání SMS zpráv při činnosti v laboratoři atd.



## Reference

- [1] BENNETT, S. (1979). A History of Control Engineering 1800-1930. London: Peter Peregrinus Ltd. pp. 47, 266. ISBN 0-86341-047-2 [cit. 20.02.2019].  
Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-pozadavky.htm> [2019-05-13]
- [2] UniPi [online]. Copyright © 2014 [cit. 22.02.2019].  
Dostupné z: <https://www.unipi.technology/cs/produkty/unipi-neuron-3> [2019-05-13]
- [3] RFID portál. RFID portál [online]. Copyright © Jelly [cit. 25.02.2019].  
Dostupné z: [https://www.rfidportal.cz/index.php?page=rfid\\_obecne](https://www.rfidportal.cz/index.php?page=rfid_obecne) [2019-05-13]
- [4] Sériový port RS-232 [online]. Copyright © [cit. 25.02.2019].  
Dostupné z: <https://www.papouch.com/cz/website/mainmenu/clanky/jak-na-to/rs232/> [2019-05-13]
- [5] SÁNDOR, Václav. Průmyslové značení výrobků. Praha, 2017. Bakalářská Práce (Bc.). České vysoké učení technické, Fakulta elektrotechnická, Katedra ekonomiky, manažerství a humanitních věd, 2017-05-25
- [6] Průmyslová sběrnice Profibus | Vývoj.HW.cz. Vývoj.HW.cz | Vše o elektronice a programování [online]. Copyright © 1997 [cit. 02.04.2019].  
Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/rozhrani/prumyslova-sbernice-profibus.html> [2019-05-13]
- [7] Nouzové zastavení stroje. Volba a použití tlačítek E-STOP | Automatizace.HW.cz. Automatizace.HW.cz | Elektronika v automatizaci [online]. Copyright © 1997 [cit. 06.04.2019].  
Dostupné z: <https://automatizace.hw.cz/bezpecnost-stroju-komponenty/nouzove-zastaveni-stroje-volba-a-pouziti-e-stop-tlacitek.html> [2019-05-13]
- [8] ŠUTKA, Jan. Zavedení revidované normy 17025 v podmínkách zkušebních laboratoří. Praha, 2018. Diplomová Práce (Ing.). České vysoké učení technické, Fakulta elektrotechnická, Katedra ekonomiky, manažerství a humanitních věd, 2017-05-25
- [9] PLC FATEK » Ladder diagramy – základy. In: PLC FATEK [online]. Copyright 2019 SEA spol. s.r.o., [cit. 06.04.2019].  
Dostupné z: <http://fatek.seapraha.cz/zaklladr/> [2019-05-13]

- [10] Function Block Diagram (FBD) Programming Tutorial - PLC Academy. In: PLC Programming & Automation Online - PLC Academy [online]. Copyright 2015 PLC Academy, [cit. 06.04.2019].  
Dostupné z: <https://www.plcademy.com/function-block-diagram-programming/> [2019-05-13]
- [11] Siemens AG. Simatic STEP-7 HELP [online]. Copyright 2007 Siemens, [cit. 06.04.2019].  
Dostupné z: [https://cache.industry.siemens.com/dl/files/056/18652056/att\\_70829/v1/S7prv54\\_e.pdf](https://cache.industry.siemens.com/dl/files/056/18652056/att_70829/v1/S7prv54_e.pdf) [2019-05-13]
- [12] Siemens AG. Datasheet 6ES7315-2EH14-0AB0 – CPU-315-DP. [online] Copyright 2007 Siemens [cit. 10.04.2019]

## Seznam obrázků a tabulek

Obr. 1	PLC od výrobce Siemens.....	4
Obr. 2	Struktura UniPi.....	6
Obr. 3	Soustava RFID.....	8
Obr. 4	Popis komunikace RS232.....	9
Obr. 5	Termotransferový tisk.....	11
Obr. 6	Příklad štítku.....	11
Obr. 7	Schéma řízení.....	12
Obr. 8	Popis kabelu ProfiBusu.....	13
Obr. 9	Přívod ze sítě a ochranné prvky.....	13
Obr. 10	Zapojení řídicí části.....	14
Obr. 11	Zapojení kabelu Cannon DB-9 k tiskárně.....	15
Obr. 12	Sestavené PLC.....	16
Obr. 13	Panel KP400 Comfort.....	16
Obr. 14	Ukázka schématu.....	17
Obr. 15	Schéma nouzového tlačítka.....	18
Obr. 16	Srovnání jazyků v závislosti na produktivitě a požadavku na paměť.....	20
Obr. 17	Ukázka jazyku LAD.....	21
Obr. 18	Ukázka jazyku FBD.....	22
Obr. 19	Ukázka jazyku STL.....	22
Obr. 20	Struktura programu PLC.....	24
Obr. 21	Rozdělení paměti PLC.....	26
Obr. 22	HW konfigurace.....	29
Obr. 23	Část datasheetu CPU.....	30
Obr. 24	Síťové propojení v TIA Portal.....	31
Obr. 25	Šablona obrazovky HMI.....	32
Obr. 26	Obrazovka přihlášení.....	33
Obr. 27	Úryvek kódu z FB1.....	34
Obr. 28	Menu.....	34
Obr. 29	Systémová funkce BLKMOV.....	35
Obr. 30	Použití funkce SFC20.....	36
Obr. 31	Ukazatel ANY a jeho struktura.....	36
Obr. 32	Funkce Logování.....	38
Obr. 33	Obrazovka tiskárny a jejího nastavení.....	39
Obr. 34	Kód ZPL pro tiskárnu.....	39
Obr. 35	Vyčítání data a času pomocí SFC1.....	40
Obr. 36	Webové rozhraní HMI.....	45
Obr. 37	Ochrana PLC.....	46

Tab. 1	Srovnání PLC .....	7
Tab. 2	Cycle time CPU pro různé operace.....	15
Tab. 3	Ukázka kusovníku .....	17
Tab. 4	Základní datové typy .....	23
Tab. 5	Komplexní datové typy .....	23
Tab. 6	Memory clock byte .....	30
Tab. 7	Paměťové oblasti .....	37

## Přílohy

Příloha č. 1	Schéma zapojení	-	ProfiCAD
Příloha č. 2	Kusovník	-	Excel
Příloha č. 3	Program PLC	-	STEP 7
Příloha č. 4	Program HMI	-	TIA Portal