

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5414-61846

# **SCADA systém produkcie v STEP7 TIA Portal**

**Diplomová práca**



**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5414-61846

# **SCADA systém produkcie v STEP7 TIA Portal**

**Diplomová práca**

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Študijný odbor: 5.2.14. automatizácia

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky

Vedúci záverečnej práce: Ing. Richard Valo, PhD.

Konzultant: Roman Slovák

**Bratislava 2016**

**Bc. Zuzana Lacková**





## ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Zuzana Lacková**  
ID študenta: 61846  
Študijný program: automatizácia a informatizácia v chémii a potravinárstve  
Študijný odbor: 5.2.14. automatizácia  
Vedúci práce: Ing. Richard Valo, PhD.  
Konzultant: Roman Slovák

Názov práce: **SCADA systém produkcie v STEP7 TIA Portal**

Špecifikácia zadania:

Práca sa bude zaoberať tvorbou SCADA systému pre automobilový priemysel.

SCADA je skratka pre "Supervisory Control And Data Acquisition", teda "dispečerské riadenie a zber dát". Obvykle sa tento pojem používa pre software, ktorý z centrálného pracoviska monitoruje priemyselné a iné technické zariadenia a procesy a umožňuje ich ovládanie.

Úlohy:

1. Oboznámenie sa s prevádzkov
2. Topológia komunikácie
3. Samotný návrh vizualizácie
4. Rozplánovanie produkcie pre zmeny s prestávkami
5. Vizualizácia prestojov
6. Vizualizácia stavu zásob medzi výrobou a zákazníkom

Rozsah práce: 50

Riešenie zadania práce od: 31. 03. 2016

Dátum odovzdania práce: 22. 05. 2016

L. S.

**Bc. Zuzana Lacková**  
študentka

**prof. Ing. Miroslav Fikar, DrSc.**  
vedúci pracoviska

**prof. Ing. Miroslav Fikar, DrSc.**  
garant študijného programu

## PodĎakovanie

Týmto by som sa chcela poĎakovať vedúcemu práce pánovi Ing. Richardovi Valovi, PhD. za odborné rady a pripomienky pri vypracovaní diplomovej práce. Ďakujem za možnosť vykonania diplomovej práci v automobilovom priemysle, za odborné rady, pripomienky, ochotu a čas konzultanta pána Romana Slováka.

## Abstrakt

V práci sa zaoberám teóriou o programovateľných logických regulátoroch (PLC), písaním programu pre PLC v Step7, teóriou o tvorbe vizualizácie v programe určenom pre tvorbu vizualizácie WinCC a ich praktickom využití v automobilovom priemysle. Opisujem topológiu komunikácie medzi programovateľným logickým regulátorom a vizualizáciou, ktorá beží v reálnom čase. Zaoberám sa konkrétnym príkladom využitia PLC a vizualizácie dát stavu produkcie v reálnom čase pri výrobe sedačiek do áut.

**Kľúčové slová:** SCADA, PLC, vizualizácia, Step 7, WinCC

## Abstract

The topic of this work are programmable logical controllers (PLC). The work involves programming of PLC in Step7, the theory of visualization creation in a program dedicated to visualization (WinCC) and its practical exploitation in automotive industry. I describe topology of communication between programmable logical controller and visualization, which runs in real time. An example case is presented, involving PLC utilization and visualization of state data in real time during production of car seats.

**Key words:** SCADA, PLC, visualization, Step 7, WinCC

# Obsah

<b>1</b>	<b>Teoretická časť</b>	<b>5</b>
1.1	Definovanie základných pojmov . . . . .	5
1.2	História PLC . . . . .	7
1.2.1	Požiadavky na PLC . . . . .	7
1.2.2	Prvé PLC . . . . .	8
1.3	PLC SIMATIC S7-300 . . . . .	9
1.3.1	Prídavné moduly S7-300 . . . . .	11
1.4	Vytváranie projektu v STEP 7 . . . . .	13
1.4.1	Vytvorenie hardvérovej konfigurácie . . . . .	14
1.4.2	Obsah projektu v STEP7 . . . . .	15
1.4.3	Tabuľka premenných . . . . .	17
1.4.4	Programovacie jazyky v STEP7 . . . . .	19
1.5	Možnosti prepojenia PLC a STEP7 . . . . .	20
1.6	Priemyselná komunikácia . . . . .	22
1.7	Spracovanie programu v PLC . . . . .	26
1.8	Vizualizácia v TIA Portal V13 - WinCC Advanced . . . . .	27
1.9	Praktické využitie PLC od firmy Siemens a vizualizácie . . . . .	29
<b>2</b>	<b>Praktická časť</b>	<b>30</b>
2.1	Definovanie cieľu práce . . . . .	30
2.2	Topológia komunikácie . . . . .	30
2.3	Vytvorenie programu v STEP 7 . . . . .	32
2.4	Konfigurácia použitého PLC . . . . .	32
2.4.1	Vytvorenie dátového bloku DB v STEP7 . . . . .	34
2.4.2	Vytvorenie funkcie FC v STEP7 . . . . .	36
2.5	Použitý polohový senzor . . . . .	38
2.6	Prepojenie PLC a WinCC . . . . .	39
2.7	Vizualizácia vo WinCC Advanced . . . . .	40
2.7.1	Definovanie Tagov . . . . .	40



2.7.2	Nastavenie rozlíšenia obrazoviek . . . . .	41
2.7.3	Prvý Screen - Basic . . . . .	42
2.7.4	Druhý Screen - PartsBoard . . . . .	44
2.7.5	Tretí Screen - Setting . . . . .	47
2.7.6	Štvrtý Screen - StopLine . . . . .	49
2.8	WinCC na riadenie . . . . .	51
2.8.1	Ďalšie prídavne funkcie . . . . .	51
2.9	Ukladanie dát do Excelu . . . . .	53
<b>3</b>	<b>Záver</b>	<b>55</b>

## Zoznam obrázkov

1.1	Základná štruktúra PLC . . . . .	7
1.2	Zľava: Dick Morley, Tom Boissevain, MODICON 084, George Schwenk, Jonas Landau [4] . . . . .	9
1.3	PLC SIMATIC S7-300 (CPU319-3 PN/DP) [3] . . . . .	10
1.4	Rozvádzač pre linku na výrobu pravých predných sedačiek . . . . .	12
1.5	Softvérový program STEP 7 . . . . .	13
1.6	Hardvérová konfigurácia . . . . .	14
1.7	Druhy blokov v STEP7 . . . . .	16
1.8	Ukážka- Tabuľka premenných . . . . .	17
1.9	Základne údajové typy [13] . . . . .	18
1.10	Zložené údajové typy [13] . . . . .	19
1.11	Vľavo STL, vpravo LAD [11] . . . . .	20
1.12	Kábel RS232 [15] . . . . .	20
1.13	Komunikačná karta CP5512 od SIEMENS [2] . . . . .	21
1.14	Komunikačný procesor CP5711 [16] . . . . .	21
1.15	NETlink adaptér . . . . .	22
1.16	Kábel PROFIBUS [8] . . . . .	23
1.17	Kábel ETHERNET [7] . . . . .	24
1.18	Kábel PROFINET [6] . . . . .	25
1.19	Činnosť PLC . . . . .	26
1.20	Prostredie WinCC Advanced . . . . .	28
1.21	Interné a externé tagy . . . . .	29
2.1	Topológia komunikácie . . . . .	31
2.2	Hardvérová konfigurácia linky pre výrobu pravých predných sedačiek . . . . .	33
2.3	Definovanie IP adresy komunikačného procesoru . . . . .	34
2.4	Dátový blok "Shift"- zabalený . . . . .	35
2.5	Výsek z dátového bloku "Shift"- rozbalený . . . . .	36
2.6	Funkcia zostrojenia ukazovateľa pre aktuálnu hodinu času . . . . .	37

2.7	Funkcia pre zápis počtu vyrobených kusov sedačiek . . . . .	37
2.8	Indukčné polohové senzory . . . . .	38
2.9	Nastavenie prepojenia WinCC s PLC . . . . .	39
2.10	Nastavenie rozlíšenia . . . . .	41
2.11	Skript na čítanie dát zo súboru . . . . .	43
2.12	Prvá obrazovka „Basic“ . . . . .	43
2.13	Výsek z funkcie pre výpočet počtu kusov, ktoré potrebujeme vyrobiť a pre výpočet zostatku . . . . .	44
2.14	Výsek z funkcie pre výpočet komulatívu za hodinu a priebežne za zmenu . . . . .	45
2.15	Funkcia pre výpočet počtu kusov, ktoré chceme vyrobiť za celú zmenu	46
2.16	Funkcia pre výpočet počtu kusov, ktoré reálne vyrobíme za celú zmenu	46
2.17	Druhá obrazovka „PartsBoard“ . . . . .	47
2.18	Tretia obrazovka „Setting“ . . . . .	48
2.19	Funkcia, ktorá aktivuje štvrtú obrazovku v prípade výskytu poruchy	49
2.20	Funkcia pre rátanie sekúnd prestoju . . . . .	50
2.21	Štvrtá obrazovka StopLine . . . . .	50
2.22	Obrazovka pre riadenie produkcie . . . . .	51
2.23	Prepínanie medzi prvou a druhou obrazovkou . . . . .	52
2.24	Reset výroby . . . . .	52
2.25	Script na zápis dát do Excelu . . . . .	53
2.26	Výsledok ukladania dát do Excelu . . . . .	54

## Zoznam skratiek a značiek

PLC- *programovateľný logický regulátor (z ang. Programmable Logic Controller)*

SCADA- *dispečerské riadenie a zber dát (z ang. Supervisory Control And Data Acquisition)*

LAD- *rebríkový diagram (z ang. Ladder diagram)*

STL- *program písaný po jednotlivých riadkoch a inštrukciach (z ang. Statement List)*

FBD- *diagram funkčných blokov (z ang. Function Block Diagram)*

DB- *dátový blok (z ang. Data Block)*

FC- *funkcia (z ang. Function)*

# Úvod

V praxi sa v priemysle čoraz častejšie vyžaduje ukladať, spracovávať a vizualizovať dáta. Tieto požiadavky prispievajú k zjednodušeniu a sprehľadneniu diania v prevádzkach. V prvých kapitolách uvediem teóriu programovateľných logických regulátorov, ktoré sa v priemysle čoraz častejšie využívajú na automatizáciu. Ďalej sa zameriam na teóriu ohľadom vizualizácie dát. V posledných kapitolách uvediem praktickú časť mojej práce, ktorá sa zaoberá vizualizáciou dát spojených so systémom produkcie v automobilovom priemysle pri výrobe sedačiek do áut. Takáto vizualizácia dát výrobného procesu prispieva k lepšiemu sprehľadneniu diania v prevádzkach. Vizualizovať môžeme takmer všetko, čo nám napadne. Či už stav zásob, počet vyrobených kusov, dĺžku trvania prestoju a rôzne iné. V mojej práci vizualizujem niekoľko dát, ktoré budú uvedené v praktickej časti tejto práce. Ďalšia výhoda vizualizovania dát je aj ich ukladanie do dokumentov ako sú napr. textové editori, Excel a iné. V práci uvediem, konkrétne ukladanie vybraných dát do Excelu. Toto ukladanie dát je dôležité hlavne pri štatistike celkovej výroby, ako napríklad, ktorá smena vyrobila koľko kusov výrobku, kedy vznikajú prestoje a podobne. Na základe zozbierania dát za určité obdobie sa dajú potom vytvoriť opatrenia, ktoré napríklad znížia čas prestojov, zvýšia produkciu a podobne. Vizualizácia dát je výborným pomocníkom pre lepšie sprehľadnenie diania, či už v priemysle alebo doma.

# 1 Teoretická časť

## 1.1 Definovanie základných pojmov

**SCADA systém-** (z ang. *Supervisory Control and Data Acquisition*) je skratka pre dispečerské riadenie a zber dát. Tento pojem sa väčšinou používa pre softvér, ktorý z centrálného pracoviska riadi a monitoruje priemyselné alebo iné technické procesy a zariadenia. Sú to napríklad priemyselné procesy ako výroba, technické procesy v distribučných sieťach (elektrina, plyn, voda), riadenie dopravných sietí (dopravná signalizácia, vetranie v tuneloch) a podobne. SCADA systémy vo všeobecnosti slúžia teda na riadenie, zber údajov, komunikáciu v sieti, prezentáciu a zobrazovanie údajov. Na zabezpečenie takýchto funkcií slúžia zariadenia ako snímače (analogové, digitálne), akčné členy, mikropočítače alebo programovateľné logické regulátory PLC, ktoré zbierajú údaje zo snímačov a posielajú príkazy na riadenie akčných členov a komunikačná sieť, ktorá zabezpečuje komunikáciu medzi jednotlivými zariadeniami.

**Vizualizácia procesov-** znamená použitie teoretických, technických, programových a komunikačných prostriedkov v priemyselnom podniku na zviditeľňovanie definovaných objektov týkajúcich sa technologických procesov a ich riadiaceho systému na podporu pri rozhodovaní v reálnom čase. [17]

**PLC-** je skratka pre programovateľný logický regulátor (z angl. *Programmable Logic Controller*). Tento pojem predstavuje prístroj, ktorý riadi určitý proces (napr. výrobnú linku v závodoch a pod.). Vykonáva sa to na základe pokynov programu, ktorý sa nachádza v pamäti PLC. Pre komunikáciu s okolím je programovateľný logický regulátor vybavený vstupnými perifériami (vstupy) na ktoré sú privádzané signály z riadeného procesu. Buď binárne signály v podobe stavov zapnuté/vypnuté (napr. snímanie polohy snímačom) alebo v podobe spojitých analogových signálov (napr. výška hladina, teplota, tlak a pod.). Na druhej strane má PLC výstupné

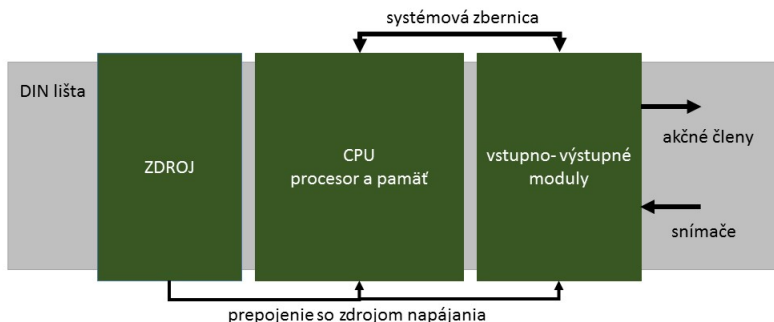
periférie (výstupy), ku ktorým sú pripojené akčné prvky riadeného systému. Tiež buď v podobe binárneho riadiaceho signálu zapnuté/vypnuté (napr. stykač motora a pod.) alebo v podobe výstupu riadiaceho signálu analógovej veličiny (napr. riadenie rýchlosti, poloha regulačného ventilu a pod.). Vstupy a výstupy spracováva centrálna procesorová jednotka CPU (riadiaca logika), ktorá na základe vstupov ovláda výstupy tak aby bola dosiahnutá minimálna odchýlka od žiadaného stavu. Programátor určuje ako bude PLC reagovať na zmenu stavov vstupných signálov, pomocou programového algoritmu (programu), ktorý rieši zadané úlohy. Tento program sa ukladá do pamäti PLC a operačný systém PLC zaistí cyklické (opakované) vykonávanie programu. Okrem klasických periférii binárnych a analógových je PLC vybavené rozhraním pre komunikáciu s programátorom alebo obsluhou (interface). Ďalšia možnosť je zapojenie PLC do siete, kde môže PLC komunikovať s ďalšími PLC alebo zariadeniami, ktoré sa nachádzajú v sieti.

**Rozdelenie PLC:** programovateľné logické regulátory môžeme z hľadiska konštrukcie rozdeliť na kompaktné a modulárne.

- **Kompaktné PLC** obsahujú všetko v jednom zariadení, teda centrálna procesorová jednotka, vstupy a výstupy sa nachádzajú v jednom zariadení.
- **Modulárne PLC** má jednotlivé komponenty rozdelené do modulov.

Základnú štruktúru PLC vidíme na obr. 1.1. Vľavo sa nachádza zdroj napájania. Ďalej je tu centrálna procesorová jednotka (CPU), v ktorej beží firmvér a užívateľský program. Pod pojmom firmvér rozumieme operačný systém PLC, ktorý zabezpečuje chod celého PLC, vykonávanie užívateľského programu a komunikáciu s ďalšími prídavnými modulmi. PLC obsahuje aj vstupno-výstupne moduly, ktoré slúžia na prístup k dátam zo snímačov. Tieto moduly je možné pripojiť k centrálnej procesorovej jednotke (CPU) prostredníctvom systémovej zbernice. Tieto moduly umožňujú zber dát z riadeného procesu a realizáciu akčných zásahov do procesu.

Všetky časti PLC sa inštalujú väčšinou na DIN lištu, ktorá sa umiestňuje do rozvážača.



Obr. 1.1: Základná štruktúra PLC

## 1.2 História PLC

### 1.2.1 Požiadavky na PLC

Prvý programovateľný logický regulátor (PLC) vyžadoval pre svoje fungovanie prostredie podobné sálovým (strediskovým) počítačom. To znamená klimatizáciu, čisté prostredie, stabilnú dodávku energie a odrušenie od vonkajších vplyvov. Na programovanie bolo nutné mať špeciálne školených programátorov, ktorý veľmi dobre poznali PLC už od samotného jadra. K praktickému využitiu programovateľných logických regulátorov (PLC) boli však potrebné podmienky skôr opačné. To znamená odolnosť proti vplyvom vonkajšieho prostredia, možnosť inštalovať ho všade



tam kde ho potrebujeme, možnosť ľahkej modifikácie pre požadované vstupy a výstupy, možnosť jednoduchých zmien aj bez dlhoročnej praxe a v neposlednom rade rýchli prenos dát.

### 1.2.2 Prvé PLC

Prvé PLC vzniklo myšlienkou náhrady reléových riadiacich systémov, ktoré mali pevnú logiku za počítačové systémy, ktoré by bolo schopné pružnejšie reagovať na zmeny vo výrobe. Teda bola požiadavka na pružnejší automatizačný systém. S touto myšlienkou prišiel na trh americký výrobca automobilov General Motors. Táto spoločnosť vyhlásila v roku 1968 súťaž na dodávku počítačového riadenia pre svoje korporácie vo Washingtonu, kde boli riešené výrobné praktiky a vznikol návrh na vytvorenie štandardu riadenia strojov. Do tejto súťaže sa prihlásili štyri spoločnosti: Information Instruments, Inc (po roku premenovaná na Allen-Bradley, teraz Rockwell Automation), Digital Equipment Corporation (DEC), Century Detroit a Bedford Associates (neskôr premenovaná na Modicon). Spoločnosti navrhli riadiace systémy podobné minipočítačom v tej dobe. Víťazom súťaže bola spoločnosť Bedford Associates a v roku 1969 bolo už vyrobené prvé PLC. Jadrom tohto PLC bol mikroprocesor (CPU), ktorý bol už prispôsobený pre priemyselné prostredie. S okolím bol prepojený cez rámy (racky) do ktorých sa inštalovali vstupno-výstupné moduly, kde každý mal 16 binárnych signálov. Celkom bolo možné k CPU pripojiť 8 rámov, každý s 8 pozíciami pre vstupno-výstupne moduly. To znamená 1024 signálov ( $8 \times 8 \times 16 = 1024$ ). Prvé PLC malo označenie 084 ako 84. projekt spoločnosti Bedford Associates, ktorá sa začala následne zaoberať výrobou, servisom a vývojom tohto produktu. Toto PLC bolo na trh uvedené po názvom MODICON (zo ang. názvu MO-dular DI-gital CON-troller) ako modulárny systém pre riadenie procesov. Jeden z prvých PLC 084 je vystavený v sídle Modiconu, North Andover, Massachusetts. Za 'otca' PLC sa pokladá vývojár Dick Morley (obr. 1.2).



Obr. 1.2: Zľava: Dick Morley, Tom Boissevain, MODICON 084, George Schwenk, Jonas Landau [4]

### 1.3 PLC SIMATIC S7-300

PLC SIMATIC S7-300 je konkrétna rada (300) PLC. Je to modulárny riadiaci systém pre široký výkonový rozsah. Ako každé iné PLC obsahuje širokú radu centrálnych procesorových jednotiek (CPU), ktoré sa líšia výkonom, osadením, počtom a druhom portov pre pripojenie ďalších zariadení. Každé CPU sa dá rozširovať o ďalšie prídavné moduly ako sú: moduly analógových signálov, moduly digitálnych vstupov a výstupov, moduly komunikačných rozhraní a ďalšie špeciálne moduly.

Ja som pri mojej práci použila PLC SIMATIC S7-300 (CPU319-3 PN/DP)(obr. 1.3). Toto CPU má integrované rozhranie PROFINET s dvomi portami. Jeho ďalšie vlastnosti sú uvedené v tabuľke (Tab. 1.3).



Obr. 1.3: PLC SIMATIC S7-300 (CPU319-3 PN/DP) [3]

Vlastnosti	CPU 319–3PN/DP
Rozmery (mm)	120x125x130
Firmvér	V3.2
Pamäť	
Pracovná pamäť	2 MB
Pokyny	680 K
Doba spracovania	
Bit operácie	0.004 $\mu s$
Word operácie	0.01 $\mu s$
Fixed-point operácie	0.01 $\mu s$
Floating-point operácie	0.04 $\mu s$
Bit pamäť/časovača/čítača	
Bit pamäť	8 192 bytes
S7 časovača/čítača	2 048/ 2048

Tab. 1.3: Vlastnosti CPU 319-3PN/DP [3]

### 1.3.1 Prídavné moduly S7-300

SIMATIC S7-300 je najviac používaný typ PLC v strojárenskom priemysle, nevyžaduje údržbu a ponúka veľa integrovaných funkcií. Ponúka taktiež širokú škálu funkčných modulov, ktoré slúžia na pridávanie ďalších funkcií do PLC. Sú to moduly ako napríklad: modul digitálnych vstupov a výstupov, modul analógových vstupov a výstupov, komunikačné moduly a podobne. Ďalej opíšem použitý napájací zdroj pre PLC a komunikačné moduly (obr. 1.4), ktoré som potrebovala pri práci.

#### Napájací zdroj SITOP

Siemens ponúka celú radu výkonných, kvalitných a cenovo dostupných napájacích zdrojov. Medzi nich patrí aj spínací napájací zdroj SITOP POWER 5. Napájacie zdroje SITOP pracujú na princípe rýchleho spínania napätia. Ich výhodou je nezávislosť výstupného napätia od zmien vstupného napätia, dodávajú presné výstupné napätie aj v prípade výkyvu napájacieho napätia. Tieto zdroje majú malé rozmery, nízku hmotnosť a kompaktné prevedenie.

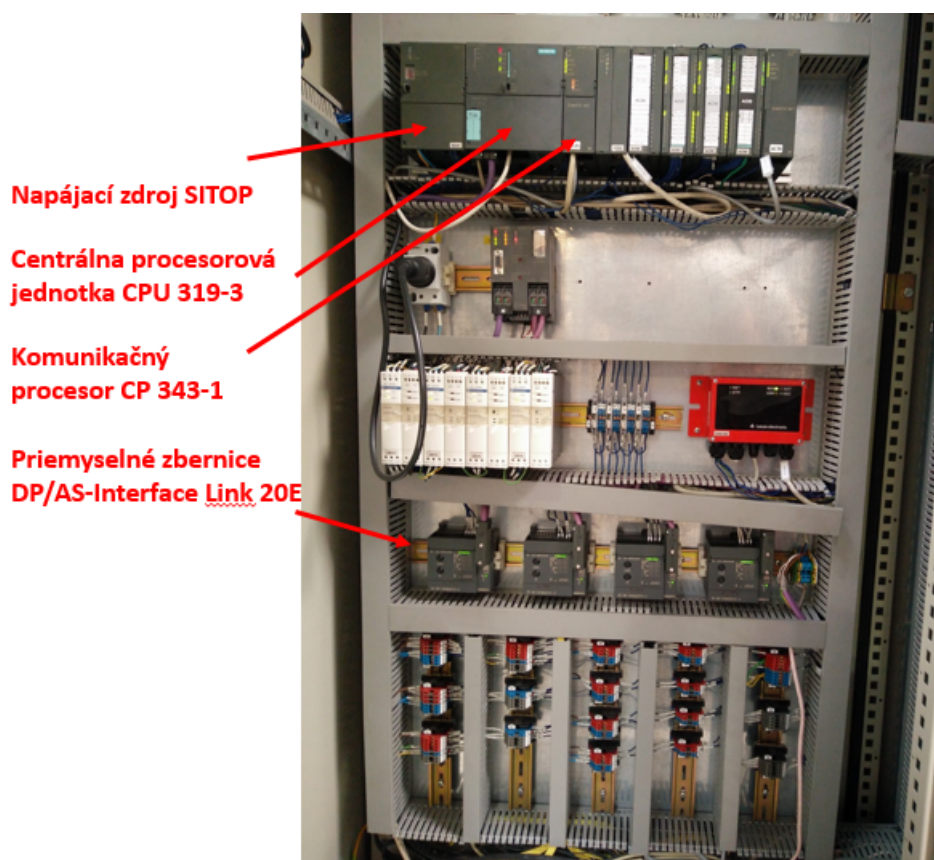
#### Komunikačný procesor CP 343-1

Komunikačný procesor je prídavný modul, ktorý slúži na komunikáciu PLC po Ethernete. Použitím tohto modulu vieme komunikovať cez FTP protokol, má HTTP server, vieme pomocou neho odosielať e-maily a cez tento modul vieme PLC spojiť so Simatic Managerom. Tak vieme pozorovať stav PLC alebo robiť úpravy aj pri väčších vzdialenostiach.

#### Komunikačný procesor DP/AS-Interface Link 20E

Ide o komunikačnú zbernicu AS-Interface (Actuator Sensor Interface), ktorá sa

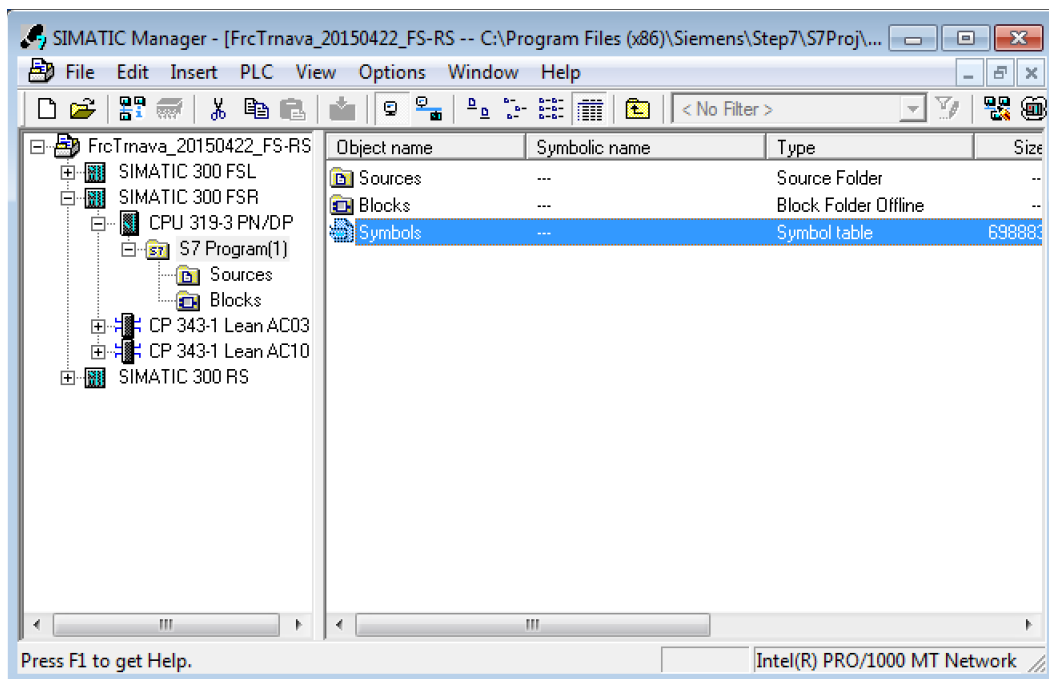
používa na zjednodušenie pripojenia veľkého počtu binárnych snímačov, senzorov, akčných prvkov a ovládacích prvkov. Na zbernici sa nachádza vždy jedno zariadenie ako master, ktoré zabezpečuje riadenie komunikácie a podľa druhu zbernice niekoľko zariadení typu slave, ktoré odiesalajú údaje na zbernicu len v prípade žiadosti zariadenia typu master. Výhodami tejto zbernice sú: jednoduchá inštalácia, jednoduché odstraňovanie porúch, jednoduché pripojenie zariadenia na zbernicu od rôznych výrobcov, vysoká spoľahlivosť.



Obr. 1.4: Rozvádzač pre linku na výrobu pravých predných sedačiek

## 1.4 Vytváranie projektu v STEP 7

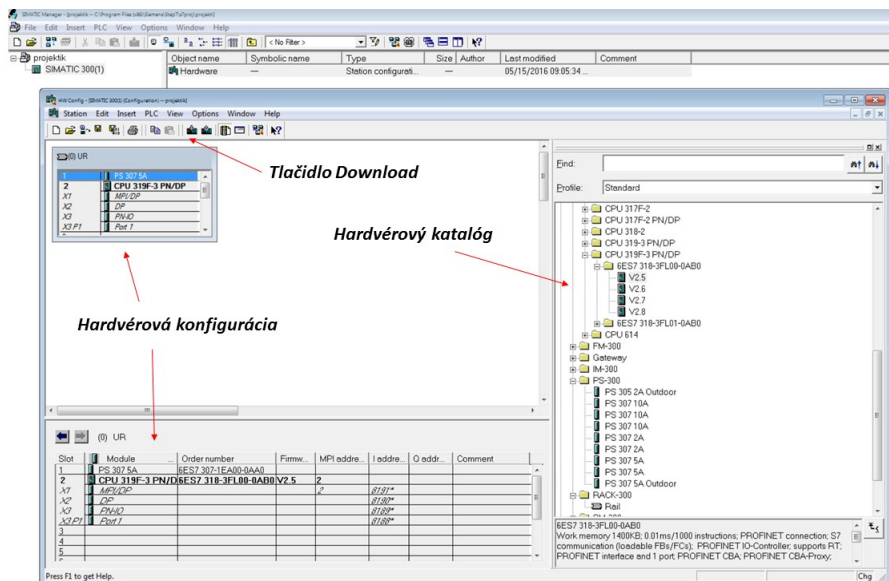
Na vytváranie vlastného programu, editáciu, správu, monitorovanie a prenos údajov do PLC a z PLC slúži softvérový nástroj STEP 7 (obr. 1.5), konkrétne prostredie SIMATIC Manager, ktorý sa inštaluje na bežný PC alebo notebook s operačným systémom Windows.



Obr. 1.5: Softvérový program STEP 7

### 1.4.1 Vytvorenie hardvérovej konfigurácie

Pri tvorbe projektu v STEP 7 je potrebné vytvoriť hardvérovú konfiguráciu. V tejto konfigurácii sa určuje jednotlivé usporiadanie modulov a priradujú sa tu adresy. Editor hardvérovej konfigurácie sa nachádza pod položkou SIMATIC 300 v časti Hardware(obr. 1.6). Nastavená hardvérová konfigurácia sa musí zhodovať so skutočným usporiadaním modulov a adresami. Centrálna procesorová jednotka porovnáva skutočnú a nastavenú konfiguráciu, ak by sa nezhodovali tak systém nenabehne. Súčasť hardvéru, ktorú chceme použiť je môžeme vyberať z hardvérového katalógu, ktorý je súčasťou STEP 7 na obr. sa nachádza vpravo. Po nastavení hardvérovej konfigurácie sa program nahráva do centrálnej procesorovej jednotky pomocou tlačidla Download na obr. 1.6.



Obr. 1.6: Hardvérová konfigurácia

### 1.4.2 Obsah projektu v STEP7

V STEP7 má každý projekt stromovú štruktúru. V prvom rade, po správnej konfigurácii nám pribudne naše CPU. Pod zložkou CPU sa nachádza zložka „S7 Program“, ktorá obsahuje ďalšie súbory pre samotné písanie programu. V prvom rade je to dôležitý súbor „Symbols“ (obr. 1.5), ktorý predstavuje tabuľku symbolických názvov premenných s ich pridelenými adresami. Ďalej je to súbor „Block“, kde sú umiestnené všetky časti programu, ktoré vytvoríme. Najskôr sa tam nachádza jeden prázdny programový blok s názvom OB1. Je to hlavný blok programu, ktorý vykonáva PLC a z neho voláme ostatné časti programu. Celý program, ktorý vykonáva PLC sa skladá z rôznych programových blokov ako sú napríklad (obr. 1.7):

#### 1. Organizačné bloky

Organizačné bloky (OB) sú rozhraním medzi užívateľským programom a CPU. OB1 je základným organizačným blokom, ktorý sa nachádza v každom programe. Obsahuje samotný cyklický program, ktorý môže volať aj iné bloky. OB nemôžu byť volané inými blokmi. Sú volané operačným systémom na isté udalosti napr. pri spustení CPU, v určený čas a pod.

#### 2. Funkcie a funkčné bloky

Funkčné bloky (FB) majú na rozdiel od funkcií (FC) pripomínanie pamäť. Znamená to, že blok lokálnych dát je priradený k funkčnému bloku. Je to tzv. instantný dátový blok. Ak teda voláme FB musíme špecifikovať číslo instantného dátového bloku, ktorý je automaticky otvorený. Instantný dátový blok sa používa na ukladanie statických premenných.

#### 3. Dátové bloky

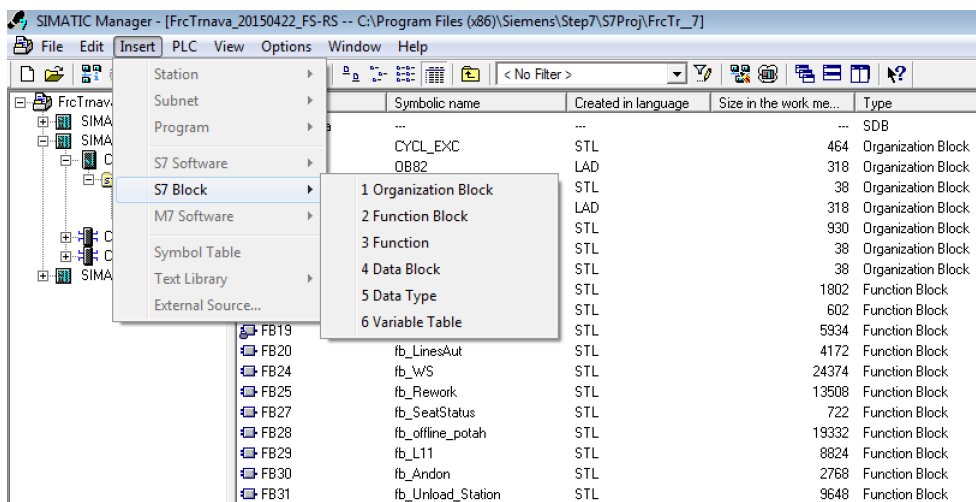
Okrem programových blokov obsahuje program aj dáta s informáciami o stavoch procesu, signáloch a podobne, ktoré sú potom spracované podľa inštrukcii programu. Je rozdiel medzi:

- globálnymi dátovými blokmi - obsahujú informácie, ku ktorým majú



prístup všetky logické bloky programu

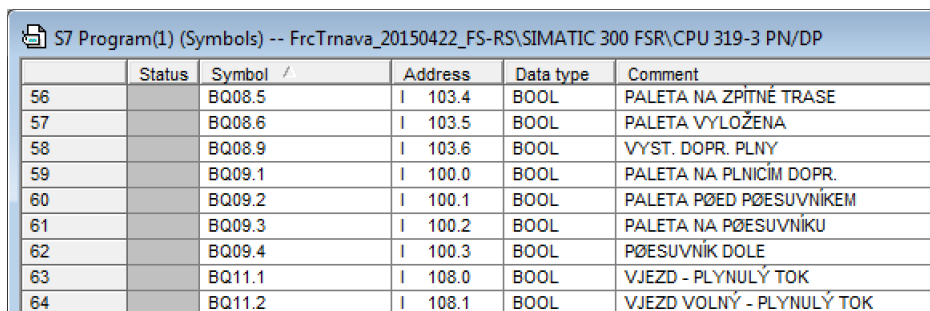
- instantné dátové bloky - sú vždy priradené konkrétnemu FB



Obr. 1.7: Druhy blokov v STEP7

### 1.4.3 Tabuľka premenných

Tabuľku premenných (obr. 1.8) nájdeme v zložke „S7 Program“ pod názvom „Symbols“. Do jednotlivých riadkov zapisujeme naše premenné v PLC, ich adresy, dátové typy a komentáre k nim.

The image shows a screenshot of the 'Symbols' table in the SIMATIC Manager software. The window title is 'S7 Program(1) (Symbols) -- FrcTrnava\_20150422\_FS-RS\SIMATIC 300 FSR\CPU 319-3 PN/DP'. The table has six columns: 'Status', 'Symbol', 'Address', 'Data type', and 'Comment'. There are 9 rows of data, numbered 56 to 64 in the first column. All data types are 'BOOL'.

	Status	Symbol	Address	Data type	Comment
56		BQ08.5	I 103.4	BOOL	PALETA NA ZPÍTNE TRASE
57		BQ08.6	I 103.5	BOOL	PALETA VYLOŽENA
58		BQ08.9	I 103.6	BOOL	VYST. DOPR. PLNY
59		BQ09.1	I 100.0	BOOL	PALETA NA PLNÍM DOPR.
60		BQ09.2	I 100.1	BOOL	PALETA PRED PŘESUVNÍKEM
61		BQ09.3	I 100.2	BOOL	PALETA NA PŘESUVNÍKU
62		BQ09.4	I 100.3	BOOL	PŘESUVNÍK DOLE
63		BQ11.1	I 108.0	BOOL	VJEZD - PLYNULÝ TOK
64		BQ11.2	I 108.1	BOOL	VJEZD VOLNÝ - PLYNULÝ TOK

Obr. 1.8: Ukážka- Tabuľka premenných

V stĺpci s názvom „Symbol“ zapisujeme názov premennej. Môže to byť ľubovoľný názov, najčastejšie sa však používa označenie, ktoré vystihuje o akú premennú ide (napr. označenie senzora). V stĺpci „Address“ sa zadáva adresa premennej podľa toho o aký druh ide napríklad vstupy (I), výstupy (Q), vnútorné premenné (M). A podľa toho o aký dátový typ ide (ten je uvedený v stĺpci „Data type“) sa priraduje číslo adresy. Teda ak je to typ BOOL zaberá len jeden bit, ak WORD 2 bajty, ak DOUBLE WORD 4 bajty. V poslednom stĺpci „Comment“ môžeme pridať komentáre k premenným. Na Obr. 1.4 vidíme, že všetky premenné sú BOOL teda dávajú hodnotu len 0 alebo 1 a sú definované ako vstupy (I). Poznáme základne, zložené a užívateľom definované údajové typy. Ich charakteristiky sú zobrazené v tabuľkách (obr. 1.9 a obr. 1.10).

Údajový typ	Deklarácia	Dĺžka bitoch	Rozsah hodnôt
Bitové typy			
Bit	BOOL	1	0,1, FALSE,TRUE
Bajt	BYTE	8	Numerický rozsah týchto bitových hodnôt nemôže byť definovaný.
Word	WORD	16	
Double Word	DWORD	32	
Znakové typy			
Znak	CHAR	8	ASCII tabuľka
Číselné typy			
Integer	INT	16	-32 768 až 32 767
Double integer	DINT	32	-2 147 483 648 až 2 147 483 647
Číslo s pohyblivou rádovou čiarkou	REAL	32	-3.402822E+38 až -1.175495E-38 1.175495E-38 až 3.402822E+38
Časové typy			
S5 Time	S5TIME	16	T#0H_0M_0S_10MS až T#2H_46M_30S <sup>2</sup>
Time	TIME	32	T#-24D_20H_31M_23S_647MS až T#+24D_20H_31M_23S_647MS <sup>1</sup>
Date	DATE	16	D#1990-01-01 až D#2168-12-31
Time of day	TIME_OF_DAY	32	TOD#0:0:0 až TOD#23:59:999

Obr. 1.9: Základne údajové typy [13]

Údajový typ	Popis
DATE_AND_TIME DT	Definuje oblasť 64 bitov (8 bytov). Uchováva čas a dátum.
STRING	Definuje oblasť pre reťazec znakov dlhý najviac 254 znakov
ARRAY	Definuje pole, skladajúce sa z iného údajového typu.
STRUCT	Definuje skupinu údajových typov v ľubovoľnej kombinácii typov.

Obr. 1.10: Zložené údajové typy [13]

#### 1.4.4 Programovacie jazyky v STEP7

STEP7 má možnosť písať program v nasledovných jazykoch [1]:

- **Function blok diagram (FBD):**

FBD je grafické zobrazenie riadiacej úlohy symbolmi s funkčnými značkami.

Na ľavej strane symbolov sa priradujú vstupy a na pravej výstupy.

- **Lader logic (LAD):**

Pomocou schematickeho zapojenia píšeme program

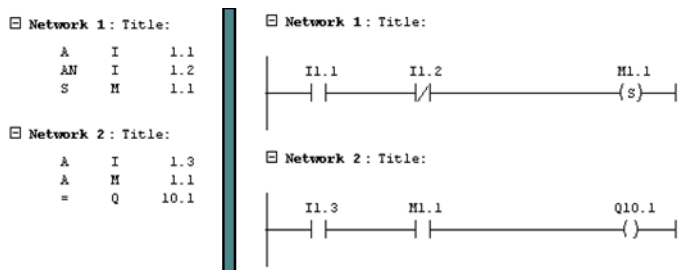
- **Statment list (STL):**

Pomocou príkazov píšeme program

Všetky programovacie jazyky možno pri písaní programu kombinovať.

Na obr. 1.11 vidno ukážku z písania programu vľavo pomocou STL (Statment list) a vpravo pomocou LAD (Lader logic), ktoré sú zhodné len napísane inými jazykmi. Každý programátor preferuje jazyk v ktorom sa mu píše najlepšie. Z praktického hľadiska sa hovorí, že programovanie v STL je rýchlejšie, lebo je to len písanie riadkov v programe. Zatiaľ čo písanie programu v LAD a FBD je pomalšie, keďže

tu vytvárame grafické zobrazenia programu, ale o to je program prehľadnejší a lepšie pochopiteľný aj pre bežného užívateľa.



Obr. 1.11: Vľavo STL, vpravo LAD [11]

## 1.5 Možnosti prepojenia PLC a STEP7

Najčastejšie používané prepojenie, hlavne pre svoju jednoduchosť a finančnú dostupnosť je MPI adaptér pre sériový port RS232 (obr. 1.12). Tieto adaptéry vyrába celá rada výrobcov vrátane SIEMENS. Postupom času sa začali objavovať aj varianty týchto adaptérov pre USB pripojenie. USB pripojenie je ale pomalšie pripojenie s PLC, čo je pre väčšie projekty nevýhodné. [2]



Obr. 1.12: Kábel RS232 [15]

Ďalším typom prepojenia je klasická komunikačná karta CP5512 od SIEMENS (obr. 1.13), ktorá využíva PCMCIA slot. Je to pripojenie, ktoré sa používalo skôr, kedy skoro nič iného neexistovalo. [2]



Obr. 1.13: Komunikačná karta CP5512 od SIEMENS [2]

Tým, že PCMCIA sloty sa z notebookov pomaly vytrácajú, tak SIEMENS svoju komunikačnú kartu CP5512 inovoval na komunikačný procesor CP5711 (obr. 1.14), ktorý má namiesto PCMCIA slotu USB port. [2]



Obr. 1.14: Komunikačný procesor CP5711 [16]

Ďalej sa medzi populárne spojenia s PLC dostávajú rôzne NETlink adaptéry (obr. 1.15), ktoré využívajú aj Ethernetové rozhranie. [2]



Obr. 1.15: NETlink adaptér

## 1.6 Priemyselná komunikácia

V modernej priemyselnej automatizácii je dôležitým prvkom komunikácia na ktorej vývoj sa kladie neustály dôraz. V posledných rokoch vzrástol dopyt po decentralizovanom riadení a vďaka zbernicovým systémom a prenikaniu Ethernetu do podnikov sa bude tento trend ďalej rozširovať. Medzi základné tri typy priemyselnej komunikácie patrí PROFIBUS, ETHERNET a PROFINET.

- **PROFIBUS**- je celosvetovo najrozšírenejší procesný zbernicový systém. Slúži na efektívnu prevádzkovú a procesnú komunikáciu s jednoduchým káblovým prepojením. Prenosová rýchlosť je 12 Mbit/s. Komunikácia je zabezpečená cez RS 485 alebo optické vlákno. Dĺžka zbernicového káblu je maximálne 1200 metrov. Profibus je bezpečná, rýchla a použiteľná aj vo výbušných prostrediach. Je určený pre nižšie až stredné komunikačné výkony.



Obr. 1.16: Kábel PROFIBUS [8]

- **ETHERNET**- priemyselný Ethernet je výkonný a široko uznávaný zbernícový systém pre dátové prenosy. Používa sa od úrovne riadenia procesov až po nadradené lokálne alebo rozsiahle siete podľa medzinárodných štandardov. Priemyselný Ethernet je vytvorený pre náročné priemyselné podmienky a disponuje výkonnou dátovou komunikáciou. Ethernet ponúka aj základné technológie pre intranet a internet. Rôzne použitie, ktoré ponúka Ethernet v oblasti kancelárskych systémov je k dispozícii aj v procesnej automatizácii a výrobe. Jeho výhodou je rýchle pripojenie, rýchla odozva, nepretržitý monitoring sieťových komponentov vďaka alarmovým hláseniam.





Obr. 1.17: Kábel ETHERNET [7]

- **PROFINET**- vychádza z Ethernetu, ktorý bol navrhnutý pre dátovú výmenu na úrovni kancelárskych aplikácií. Priemyselná komunikácia však vyžadovala vyššiu úroveň, napríklad schopnosť práce v reálnom čase, rýchle inštalačné metódy, integráciu distribuovanej prístrojovej techniky a podobne. Týmto požiadavkám vyhovuje práve komunikačný štandard PROFINET, ktorého tvorcom je organizácia Profibus & Profinet International. Využíva osvedčené štandardy IT ako napríklad TCP/IP a je schopný komunikácie v reálnom čase. Veľkou výhodou Profinetu je vzdialená údržba, diagnostika po sieti a vysoký komunikačný výkon.



Obr. 1.18: Kábel PROFINET [6]

V roku 2004 vyjadrilo podporu systému Profinet združenie nemeckých výrobcov automobilov AIDA (Automatisierungs Initiative Deutscher Automobilhersteller). Členmi AIDA sú: Audi, BMW, Daimler, Porsche a Volkswagen. Toto združenie vybralo Profinet ako systém priemyselného Ethernetu, ktorý bude používať vo svojich automatizovaných prevádzkach. Snahou AIDA bolo podporiť vývoj jednotného systému, ktorý by bol neutrálny vzhľadom k výrobcovi a užívateľovi, bezpečný a využíval by štandardné vývojové nástroje. Toto všetko Profinet spĺňa a od tej doby organizácia Profibus & Profinet International neustále reaguje na požiadavky združenia AIDA.

## 1.7 Spracovanie programu v PLC

PLC je procesor, ktorý spracováva program riadok po riadku, inštrukciu po inštrukcii. Prakticky to funguje tak, že máme dva akumulátory (pamäťové priestory pre 4 byty, teda 1 double word). A napríklad pri vykonaní príkazu na násobenie dvoch čísel, sa prvé číslo nahrá do prvého akumulátora. Potom sa zoberie druhé číslo a nahrá sa do prvého akumulátora, predtým sa ale prvé číslo presunulo do druhého akumulátora. Vždy teda PLC najprv zoberie obsah prvého akumulátora a presunie ho do druhého akumulátora aby bol prvý akumulátor uvoľnený pre premennú. V treťom kroku sa vykoná násobenie a výsledok násobenia sa uloží do prvého akumulátora. Procesor spracováva program cyklicky neustále dokola po jednotlivých inštrukciách (obr. 1.19). Najprv sa prečítajú hodnoty všetkých PLC vstupov. Ďalej náš program vykoná príkazy po jednotlivých inštrukciách a výsledky zapisuje do tabuľky výstupov.



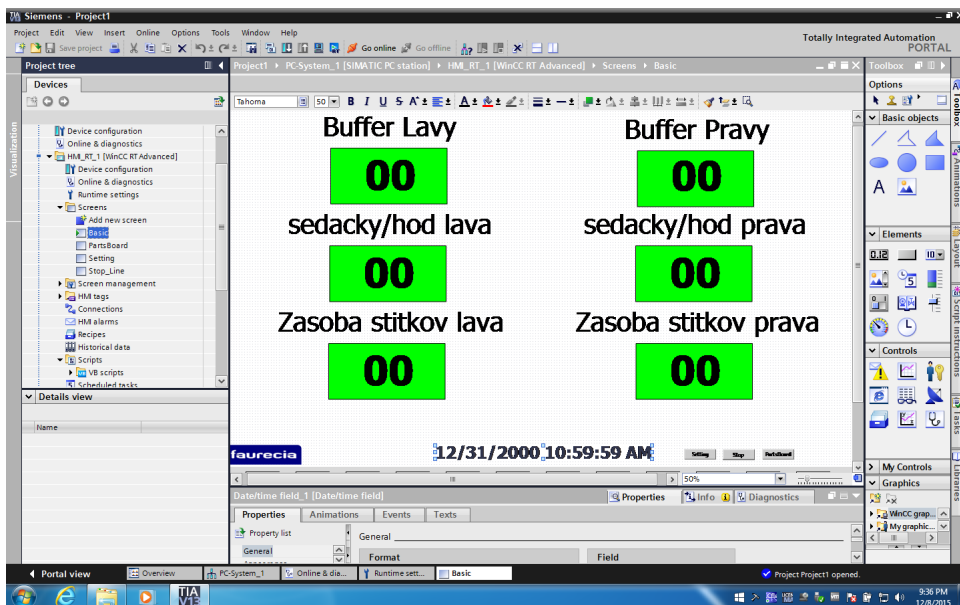
Obr. 1.19: Činnosť PLC

## 1.8 Vizualizácia v TIA Portal V13 - WinCC Advanced

Prostredie **TIA Portal** (z ang. *Totally Integrated Automation Portal*) by sme v preklade mohli preložiť ako úplný integrovaný automatizačný portál. Vývoj tohto prostredia predstavuje významný historický pokrok v priemyselnej automatizácii. TIA Portal predstavuje komplexné softvérové vývojové prostredie v ktorom je integrovaný vývoj užívateľských aplikácií pre PLC a decentralne periférie, projektovanie HMI panelov, vizualizácia SCADA systémov, sieťové komponenty, komunikačné prvky a konfigurácia systémov. Toto všetko je prístupné z jednej jedinej plochy. Znamená to zrýchlenie tvorby aplikácií, vyššiu konzistenciu dát celého projektu, jednoduchú štruktúru programu, ktorú je ľahké meniť alebo dopĺňať v bežnom chode. Tým pádom je celá štruktúra programu lepšie pochopiteľná a vyhovujúca aj pre bežného používateľa, údržbu a servis automatizačnej techniky.

Prostredie pre tvorbu vizualizácie **WinCC Advanced** (obr. 1.20) je zložené z hlavného menu, ktoré sa nachádza na titulnej lište. V strede je náhľad na HMI obrazovky alebo iné obrazovky pre ktoré vytvárame vizualizáciu. Vľavo sa nachádza okno, ktoré má stromovú štruktúru. Nachádzajú sa v ňom Screeny (obrazovky), ktoré môžeme pridávať aj premenovať. Ďalej sa tu nachádza komunikácia v ktorej sa definujú Tagy, a prepojenie s PLC. Taktiež sa tu v skriptoch dajú vytvárať rôzne funkcie. Na pravej strane sa nachádza okno s nástrojmi pre vkladanie objektov ako sú napríklad texty, tlačidlá, vstupno-výstupne polia a iné. Stromová štruktúra programu je veľmi výhodná, zabezpečuje prehľadné roztriedenie všetkých použitých činností a častí programu (napr. použitú komunikáciu s PLC, Tagy, grafiku a iné).

Tie premenné z PLC, ktoré chceme využiť vo vizualizácii musíme vložiť do Tagov (obr. 1.21).



Obr. 1.20: Prostredie WinCC Advanced

Pod pojmom Tag si môžeme predstaviť niečo ako virtuálny dátový kanál, cez ktorý prechádzajú dáta. Jeden „koniec kanálu“ je pripojený na určitú pamäťovú adresu (slúži ako zásobník pre dáta) a druhý „koniec kanálu“ sprístupňuje tieto dáta užívateľovi. Tagy delíme na dve skupiny [1]:

- **interné** - používajú sa pre interne (vnútorne) definované premenné
- **externé** - používajú sa na komunikáciu s PLC

Jednotlivým obrazovkám môžeme vo vlastnostiach priradiť svoj názov, farebné pozadie a iné vlastnosti. Je to šablóna na ktorú umiestňujeme všetko čo chceme v našej vizualizácii zobraziť. Na obrazovky môžeme umiestňovať rôzne objekty. Napríklad text umiestnime pretiahnutím textového poľa (TextField), ktoré sa nachádza na pravej strane projektu. V jeho vlastnostiach sa nachádzajú ďalšie rozširujúce mo-

HMI tags				
	Name ▲	Tag table	Data type	Connection
	Line_in_Stop_PL	Default tag table	Bool	Lava_Linka
	Line_in_Stop_PR	Default tag table	Bool	Prava_Linka
	Line_in_Stop_R	Default tag table	Bool	Prava_Linka
	Motor_Run_Lava	Default tag table	Bool	Lava_Linka
	Motor_Run_prava	Default tag table	Bool	Prava_Linka
	Nocna_Real	Default tag table	Int	<Internal tag>
	Nocna_Target	Default tag table	Int	<Internal tag>
	Offset	Default tag table	Int	<Internal tag>
	Ope/Hod_L	Default tag table	Int	Lava_Linka
	Ope/Hod_P	Default tag table	Int	Prava_Linka
	PL1_Stop	Default tag table	Bool	Lava_Linka
	PL2_Stop	Stop_Line	Bool	Lava_Linka
	PL3_Stop	Stop_Line	Bool	Lava_Linka

Obr. 1.21: Interné a externé tagy

žnosti nastavenia ako je farba písma, pozadia, animácie a podobne. Ďalej môžeme vkladať vstupno-výstupne polia (I/O Field), ktoré spájame s Tagmi, slúžia väčšinou na zobrazenie určitej hodnoty z PLC. Ďalej sú to tlačidlá (Button) a iné. Všetky objekty vkladáme rovnako ako bolo uvedené pri textovom poli a taktiež vo vlastnostiach každého objektu vieme nastaviť rozširujúce vlastnosti nastavenia objektu.

## 1.9 Praktické využitie PLC od firmy Siemens a vizualizácie

Programovateľné logické regulátory sa v spojený s vizualizáciou používajú už v takmer každom priemysle, či už chemickom (napr. vizualizácia stavu teploty v chemickom reaktore), automobilovom (napr. priebeh činnosti montáže), alebo vo výrobe (napr. vizualizácia výrobnnej linky), ale aj v inteligentných domoch (napr. vizualizácia svietidiel v domácnostiach). Samotná vizualizácia nemusí spĺňať len funkciu zobrazovania dát, ale môže mať aj funkciu zadávania dát. To znamená, že ak máme napríklad dotykový panel na ktorom máme naprogramované, že stlačením určitého tlačidla sa rozsvieti dané svetlo, tak po jeho stlačení sa svetlo rozsvieti. Dnes sa už dá vizualizácia dát použiť takmer všade.

## 2 Praktická časť

### 2.1 Definovanie cieľu práce

Základným cieľom mojej práce je navrhnuť a vytvoriť vizualizáciu systému produkcie v automobilovej prevádzke pri výrobe sedačiek do áut. Vizualizácia má obsahovať rôzne funkcie pre výpočet produkcie, objekty a iné, ktorým sa budem venovať postupne.

Výroba je zložená z troch výrobných liniek a jednej expedičnej linky. Pre každú výrobnú linku je navrhnutá vizualizácia v podstate rovnaká, líši sa len zobrazovanými číslami, lebo sa vzťahuje na danú výrobnú linku. Preto v práci uvediem spôsob tvorby vizualizácie pre pravú výrobnú linku. Pre zvyšné výrobné linky je rovnaký, líši sa len Tagmi, ktorými sú jednotlivé PLC a WinCC prepojené.

### 2.2 Topológia komunikácie

Všetky zariadenia (PLC a WinCC) na obr. 2.1 sú pripojené na sieť pomocou Ethernetu, kde každému zariadeniu prislúcha vlastná IP adresa. Toto prepojenie zabezpečuje komunikáciu medzi jednotlivými zariadeniami.

Výroba je zložená zo štyroch liniek:

1. **Ľavá linka-** je linka na ktorej sa vyrábajú ľavé predné sedačky do áut
2. **Pravá linka-** je linka na ktorej sa vyrábajú pravé predné sedačky do áut
3. **Zadná linka-** je linka na ktorej sa vyrábajú zadné sedačky do áut
4. **Expedičná linka-** slúži na expedíciu hotových sedačiek do kamiónov



Obr. 2.1: Topológia komunikácie

Každá linka má vlastný rozvádzač, v ktorom je na DIN lište umiestnené PLC, ktoré zabezpečuje riadenie výroby prislúchajúcej linky na jednotlivých staniciach. Nad linkami, kde sa vyrábajú sedačky sú umiestnené tri televízory. Každý televízor zobrazuje stav produkcie pre linku nad ktorou je umiestnený. Každý televízor je prepojený s vlastným počítačom, na ktorom beží operačný systém Windows v ktorom je nainštalovaný WinCC Runtime. Pomocou tohto WinCC Runtime je možné spúšťať vizualizáciu na PC. Ďalej sa tu nachádza ešte jedno WinCC, ktoré slúži na riadenie produkcie. Nachádza na PC majstra výroby, ktorý zadáva vstupné hodnoty ako sú rýchlosť akou má ísť linka, Takt Time a podobne.



## 2.3 Vytvorenie programu v STEP 7

Pri písaní programu som kládla dôraz na to, aby bol program napísaný v STEP 7 čo najkratší a všetko čo sa dá aby sa naprogramovalo cez funkcie vo WinCC Advanced. Je to kvôli tomu, že PLC v ktorom sa vykonáva program napísaný v STEP 7 má obmedzenú pracovnú pamäť (2000 KB), retenčnú (700 KB) a vizualizácia vytvorená vo WinCC Advanced beží na počítači, ktorý má v podstate „neobmedzenú pamäť“, lebo do neho vieme pridávať harddisky.

## 2.4 Konfigurácia použitého PLC

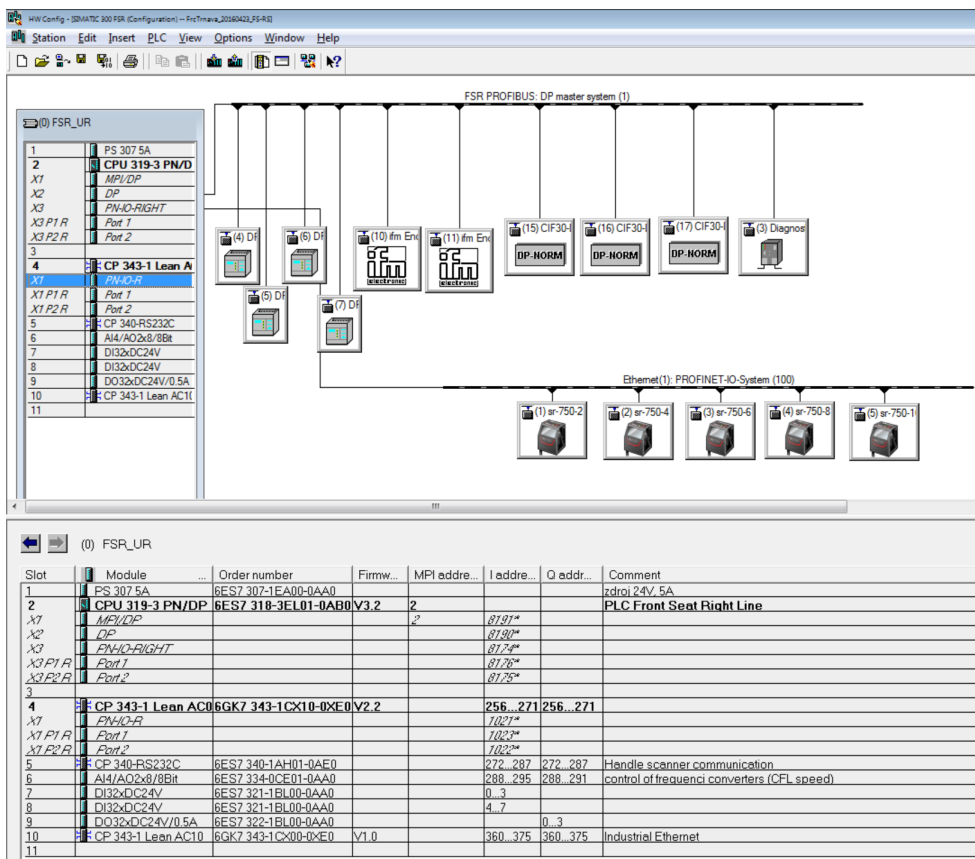
Na obr. 2.2 môžeme vidieť hardvérovú konfiguráciu PLC pre linku na ktorej sa vyrábajú pravé predné sedačky. Hardvérová konfigurácia je vytvorená z:

- zdroj napájania (PS 307 5A)
- centrálna procesorová jednotka (CPU 319-3)
- komunikačný procesor (CP 343-1)

Centrálna procesorová jednotka (CPU 319-3) disponuje s jedným portom pre Profibus a s dvomi portami pre Profinet. Na obr. 2.2 môžeme vidieť, že na Profibuse sa nachádzajú zariadenia:

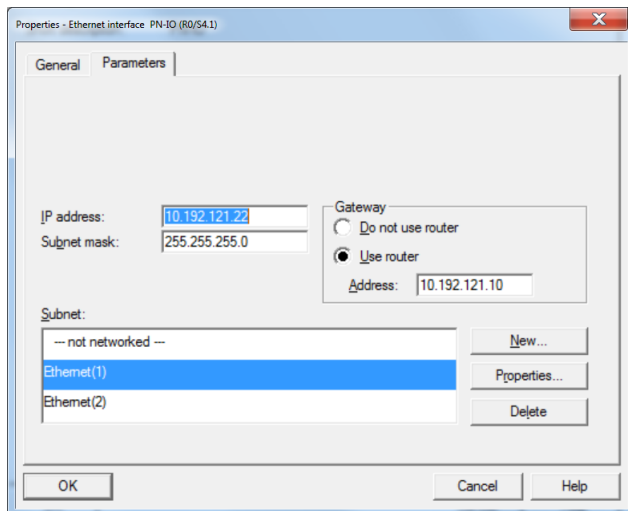
- AS-Interface- konkrétne DP/AS-i Link 20E, je to zbernicový systém pre úroveň akčných členov a senzorov
- Enkódery- sú to snímače polohy motora, slúžia na snímanie rýchlosti otáčok
- Elektrické testery- slúžia na testovanie sedačiek do áut
- Diagnostic repeater- slúži na diagnostiku siete Profibus

Na Profinete sa nachádzajú iba čítačky (skenery), ktoré sa používajú na skenovanie štítkov.



Obr. 2.2: Hardvérová konfigurácia linky pre výrobu pravých predných sedačiek

Komunikačný procesor (CP 343-1) používam na komunikáciu so všetkými WinCC bežiacich na počítačoch. Tu je potrebné zadať vo vlastnostiach IP adresu (obr. 2.3), prostredníctvom ktorej potom vytvoríme spojenie s WinCC.



Obr. 2.3: Definovanie IP adresy komunikačného procesoru

#### 2.4.1 Vytvorenie dátového bloku DB v STEP7

Dátový blok (DB) som vytvorila tak, že v STEP7 som ho vložila do programu, pomenovala som ho ("Shift") ako zmena. Ďalej som definovala premenné (obr. 2.4), ktorých hodnoty chceme ukladať. Teda vytvorila som zoznam (ARRAY[0..23]) od 0 do 23, lebo potrebujem ukladať hodnoty premenných v každej hodine. Premenné, ktoré sa môžu zapisovať každú hodinu sú počet pracovných minút v hodine (*Min\_in\_hour*), cieľ výroby (*Target\_make*), reálne vyrobené (*Real\_make*), rozdiel za hodinu (*Dif\_Hour*), rozdiel za zmenu (*Dif\_Shift*). A ďalšie premenné ako čistý pracovný čas (*Clean\_Work\_T*), sedadlá za deň (*Seat\_Per\_Day*), sedadlá za hodinu (*Seat\_Per\_Hour*), zásoba sedadiel (*Seat\_Buffer*). Do DB vieme zapisovať určené hodnoty tým, že si vytvoríme funkcie (Kap. 2.2.2). Niektoré premenné som nakoniec nevyužila, lebo som ich vedela vytvoriť vo WinCC a tým som ušetrila pamäť

PLC. Na obr.2.5 vidíme výsek dátového bloku z obr. 2.4, ktorý je rozbalený. Vytvorenie zoznamu ARRAY sa používa napríklad v takých prípadoch, keď chceme ukladať niektoré dáta každú hodinu. Zjednodušene povedané nemusíme vypisovať 24 premenných (pre každú hodinu 1 premenná), ale stačí vytvoriť len jeden zoznam ARRAY. V jazyku STL zapisujeme dáta do dátového bloku DB príkazom napríklad L 45 (z *ang. Load*) načítať číslo 45 a následne príkazom T DB1.DBW4 (z *ang. Transfer*) sa táto hodnota prenesie do dátového bloku. V prípade na obr. 2.5 by to znamenalo, že do nášho dátového bloku DB1 sa na adresu 4.0 do premennej *Hour[0].Real\_make* zapísalo číslo 45 typu WORD.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Hour	ARRAY[0..23]		
+0.0		STRUCT		
+0.0	Min_in_hour	WORD	W#16#0	
+2.0	Target_make	WORD	W#16#0	
+4.0	Real_make	WORD	W#16#0	
+6.0	Dif_Hour	WORD	W#16#0	
+8.0	Dif_Shift	WORD	W#16#0	
=10.0		END_STRUCT		
+240.0	Clean_Work_T	WORD	W#16#0	
+242.0	Seat_Per_Day	WORD	W#16#0	
+244.0	Seat_Per_Hour	WORD	W#16#0	
+246.0	Seat_Buffer	WORD	W#16#0	
=248.0		END_STRUCT		

Obr. 2.4: Dátový blok "Shift"- zabalený

Address	Name	Type	Initial value	Actual value	Comment
0.0	Hour[0].Min_in_hour	WORD	W#16#0	W#16#0	
2.0	Hour[0].Target_make	WORD	W#16#0	W#16#0	
4.0	Hour[0].Real_make	WORD	W#16#0	W#16#0	
6.0	Hour[0].Dif_Hour	WORD	W#16#0	W#16#0	
8.0	Hour[0].Dif_Shift	WORD	W#16#0	W#16#0	
10.0	Hour[1].Min_in_hour	WORD	W#16#0	W#16#0	
12.0	Hour[1].Target_make	WORD	W#16#0	W#16#0	
14.0	Hour[1].Real_make	WORD	W#16#0	W#16#0	
16.0	Hour[1].Dif_Hour	WORD	W#16#0	W#16#0	
18.0	Hour[1].Dif_Shift	WORD	W#16#0	W#16#0	
20.0	Hour[2].Min_in_hour	WORD	W#16#0	W#16#0	
22.0	Hour[2].Target_make	WORD	W#16#0	W#16#0	
24.0	Hour[2].Real_make	WORD	W#16#0	W#16#0	
26.0	Hour[2].Dif_Hour	WORD	W#16#0	W#16#0	
28.0	Hour[2].Dif_Shift	WORD	W#16#0	W#16#0	
30.0	Hour[3].Min_in_hour	WORD	W#16#0	W#16#0	
32.0	Hour[3].Target_make	WORD	W#16#0	W#16#0	
34.0	Hour[3].Real_make	WORD	W#16#0	W#16#0	
36.0	Hour[3].Dif_Hour	WORD	W#16#0	W#16#0	
38.0	Hour[3].Dif_Shift	WORD	W#16#0	W#16#0	
40.0	Hour[4].Min_in_hour	WORD	W#16#0	W#16#0	

Obr. 2.5: Výsek z dátového bloku "Shift"- rozbalený

## 2.4.2 Vytvorenie funkcie FC v STEP7

Funkciu som vytvorila pomocou bloku FC, kde prvá časť sa zaoberá zostrojením ukazovateľa (pointra) pre aktuálnu hodinu času. A druhá časť zabezpečuje zapisovanie počtu vyrobených kusov.

Na obr. 2.6 vidíme prvú časť funkcie, kde prvý riadok načíta do AR1 (pamäť) aktuálnu hodinu času. V druhom riadku sa do AR2 (pamäť) načíta číslo 10. Preto číslo 10, aby sme sa v DB vždy po vynásobení aktuálnej hodiny desiatkov a premenením decimálnej hodnoty na pointrovú dostali na adresu premennej prislúchajúcej tejto hodine. V treťom riadku tieto dve hodnoty teda vynásobíme. V štvrtom riadku sa pomocou príkazu SLD (*z ang. Shift Left Double Word*) premení decimálna hodnota na pointrovú hodnotu. Ďalej sa táto hodnota načíta do AR1 (pamäť) a nakoniec sa zapíše do premennej *P\_Hour*.

```

L      "d_Date_Time".CPU_Date_Time_INT.Hour
L      10
*I
SLD    3
LAR1
T      #P_Hour

```

Obr. 2.6: Funkcia zostrojenia ukazovateľa pre aktuálnu hodinu času

Na obr. 2.7 vidíme druhú časť funkcie. Tu sa v prvých troch riadkoch vykonáva logická operácia AND. Teda ak sú aktívne obidva senzory tak sa zoberie nábežná hrana (lebo sedačka môže byť na senzore dlhšie a my chceme len počet vyrobených), otvorí sa DB „Shift“, do AR2 načítam premennú *P\_Hour*, ktorá slúži ako ukazovateľ (pointer). Načítam aktuálnu hodnotu počtu reálne vyrobených sedačiek, pripočítam jednotku a prepíšem počet reálne vyrobených sedačiek. V dátovom bloku na adrese 246 sa nachádza premenná „*Seat\_Buffer*“ do ktorej tiež zapisujeme počet vyrobených kusov. A ak logická operácia AND nie je splnená, tak príkazom JCN (z ang. *Jump if RLO = 0*) preskočíme vykonanie pripočítania vyrobenej sedačky.

```

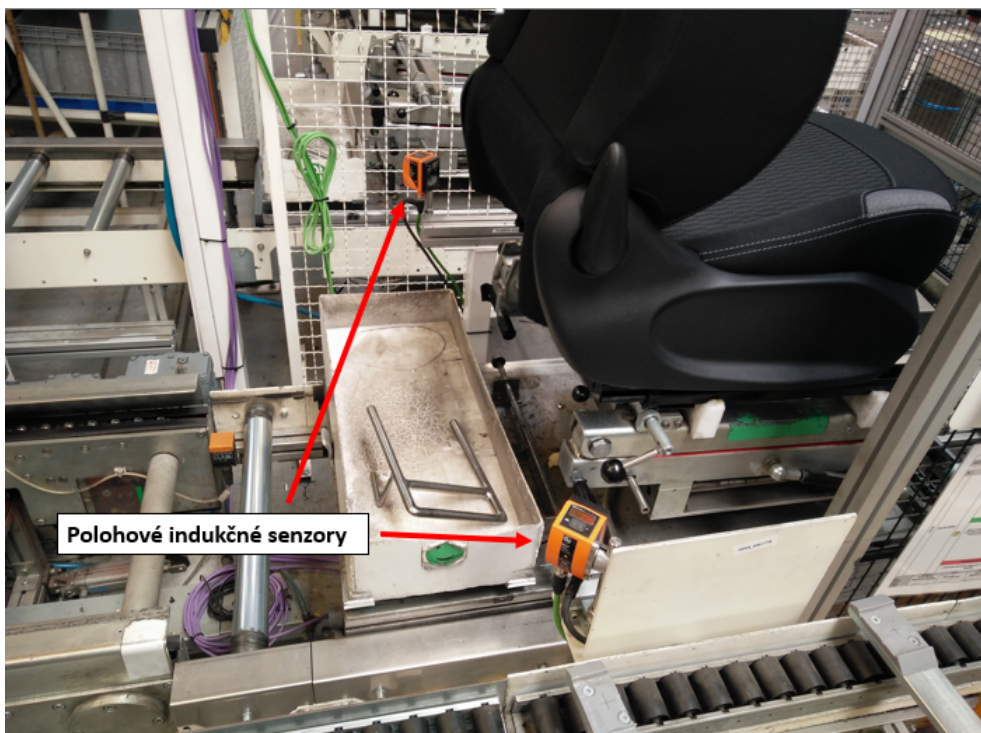
A      "BQ22.1b"
FP      M      319.0
A      "BQ21.2"
JCN     Piec
OPN     "Shift"
LAR2    #P_Hour
// Per Hour
L      DBW [AR2,P#4.0]
+      1
T      DBW [AR2,P#4.0]
// Buffer
L      DBW 246
+      1
T      DBW 246
Piec: NOP 0

```

Obr. 2.7: Funkcia pre zápis počtu vyrobených kusov sedačiek

## 2.5 Použitý polohový senzor

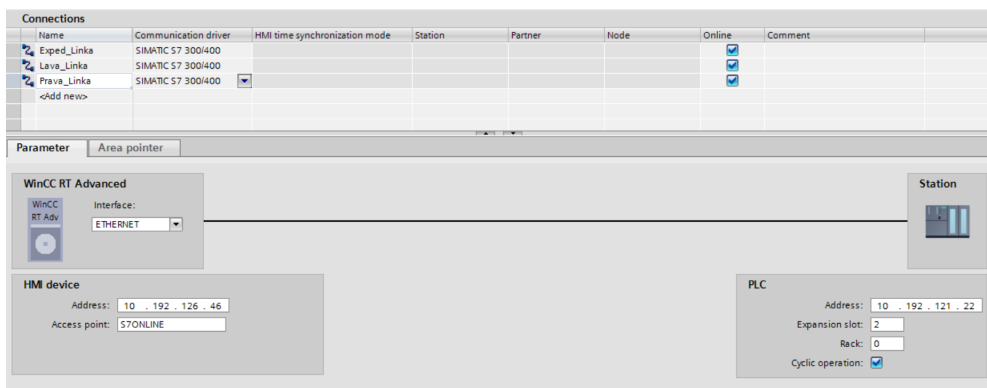
V predchádzajúcej kapitole 2.4.2 som spomínala senzory, ktoré detekujú prítomnosť sedačky. Sú to indukčné polohové senzory od firmy ifm (obr. 2.8), konkrétne typ IM513A. Ich spínacia vzdialenosť je 30 mm. Sú odolné teplotám od  $-20^{\circ}\text{C}$  až po  $60^{\circ}\text{C}$ . Ich IP krytie je IP 65, IP 67 čo znamená, že sú veľmi dobre odolné voči prachu a vode. Ich výstupnou funkciou je spínanie/rozopínanie. To znamená, že sú to digitálne senzory, ktoré nám dávajú hodnotu buď 0 alebo 1. Tieto senzory sú zapojené do AS-i zbernice. V hardvérovej konfigurácii im vieme natvrdo zadať adresu alebo ak sme v online režime tak im ju program priradí sám.



Obr. 2.8: Indukčné polohové senzory

## 2.6 Prepojenie PLC a WinCC

Programovateľné logické regulátory (PLC) komunikujú s jednotlivými vizualizáciami (WinCC) po Ethernete. Na to aby bola táto komunikácia zabezpečená treba nastaviť v TIA Portali prepojenie, ktoré vidíme na obr. 2.9. Jednotlivým WinCC sú pridelené IP adresy a pomocou Ethernetu prepojíme s PLC, ktorých IP adresu tiež nastavíme (je to IP adresa, ktorú sme nastavovali pre komunikačný procesor CP 343-1). Pomocou tohoto procesoru môže potom WinCC komunikovať s PLC. Na obr. 2.9 vidíme, že je nastavené prepojenie WinCC s expedičnou, ľavou a pravou linkou. Pod týmito prepojeniami vidíme podrobné nastavenie prepojenia WinCC s PLC z pravej linky (IP adresa z obr. 2.3 sa musí a teda aj zhoduje s IP adresou na obr. 2.9).



Obr. 2.9: Nastavenie prepojenia WinCC s PLC



## 2.7 Vizualizácia vo WinCC Advanced

### 2.7.1 Definovanie Tagov

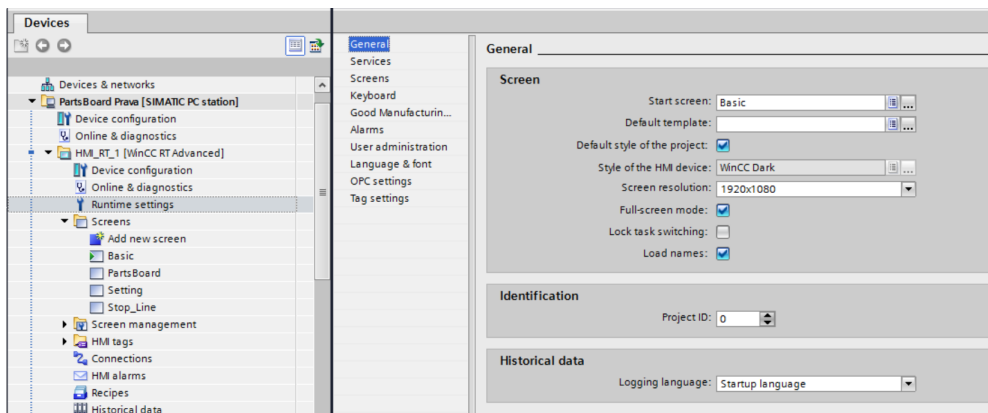
Tagy, ktoré používam vo vizualizácii sa delia na 2 skupiny. Interné tagy, ktoré sa nachádzajú len vo vizualizácii a externé tagy, ktorými komunikujeme s PLC. Hlavné Tagy, ktoré používam sú:

- „Buffer\_L“ - počet sedačiek, ktoré sa nachádzajú v buffri na ľavej linke
- „Buffer\_R“ - počet sedačiek, ktoré sa nachádzajú v buffri na pravej linke
- „Sed/Hod\_R“ - rýchlosť výroby sedačiek za hodinu na pravej linke
- „Sed/Hod\_L“ - rýchlosť výroby sedačiek za hodinu na ľavej linke
- „HodPlan\_6“ - počet sedačiek, ktoré máme vyrobiť za danú hodinu
- „Hod\_6“ - počet sedačiek, ktoré sme reálne vyrobili v danej hodine
- „Kom\_H\_6“ - komulácia sedačiek v danej hodine
- „Kom\_S\_6“ - komulácia sedačiek počas celého času
- „Ranna\_Target“ - počet sedačiek, ktoré máme vyrobiť za rannú zmenu
- „Poobedna\_Target“ - počet sedačiek, ktoré máme vyrobiť za poobednú zmenu
- „Nocna\_Target“ - počet sedačiek, ktoré máme vyrobiť za nočnú zmenu
- „Ranna\_Real“ - počet sedačiek, ktoré sme reálne vyrobili za rannú zmenu
- „Poobedna\_Real“ - počet sedačiek, ktoré sme reálne vyrobili za poobednú zmenu
- „Nocna\_Real“ - počet sedačiek, ktoré sme reálne vyrobili za nočnú zmenu
- „6H“ - predstavuje nevýrobný čas v minútach (prestávky) v danej hodine
- „6min“ - predstavuje výrobný čas v minútach v danej hodine

- „zostatok\_6“ - sekundy, ktoré sa prenášajú na vyrobenie poslednej sedačky do nasledujúcej hodiny
- „Buffer\_Limit\_L“ - limit pre buffer ľavej linky
- „Buffer\_Limit\_R“ - limit pre buffer pravej linky
- „Prestoj\_L“ - počet sekúnd, koľko je ľavá linka zastavená
- „Prestoj\_R“ - počet sekúnd, koľko je pravá linka zastavená
- „Motor\_Run\_ľava“ - určuje či je ľavá linka v pohybe alebo nie
- „Motor\_Run\_prava“ - určuje či je pravá linka v pohybe alebo nie

## 2.7.2 Nastavenie rozlíšenia obrazoviek

Pred samotným návrhom vizualizácie je potrebné nastaviť rozlíšenie obrazoviek vo vizualizáciách, aby sa rozlíšenie zariadenia na ktorom bude vizualizácia spustená zhodovalo s rozlíšením vizualizácie. Toto rozlíšenie sa nastavuje v nastaveniach Runtimu, ktoré vidíme na obr. 2.10.



Obr. 2.10: Nastavenie rozlíšenia

### 2.7.3 Prvý Screen - Basic

Prvá obrazovka (obr. 2.12) s názvom „Basic“ obsahuje objekty typu Text field (všetky textové polia), Rectangle (všetky zelené obdĺžniky), ktoré majú navyše pridanú dynamickú zmenu farby, táto zmena farby je napojená na Tag, ktorý určuje minimálny limit pre Buffer sedačiek a ak je táto hodnota menšia tak obdĺžnik má červenú farbu. Nachádzajú sa tu aj I/O field objekty, ktorými pomocou Tagov zobrazujeme určité čísla z PLC napríklad Buffer pre pravú linku, Buffer pre ľavú linku, rýchlosť výroby sedačiek a podobne. Zásoba štítkov sa číta zo súboru pomocou funkcie na obr. 2.11. Táto funkcia otvára súbor v ktorom sú v každom riadku tri čísla oddelené čiarkov. Ja čítam len prvé dve čísla, prvé číslo reprezentuje zásobu štítkov pre ľavé sedačky a druhé číslo reprezentuje zásobu štítkov pre pravé sedačky. Tieto čísla sú potom zapísané do premenných *stitkyL* a *stitkyP*. Prostredníctvom I/O field objektov sú hodnoty týchto premenných zobrazené na obrazovke.

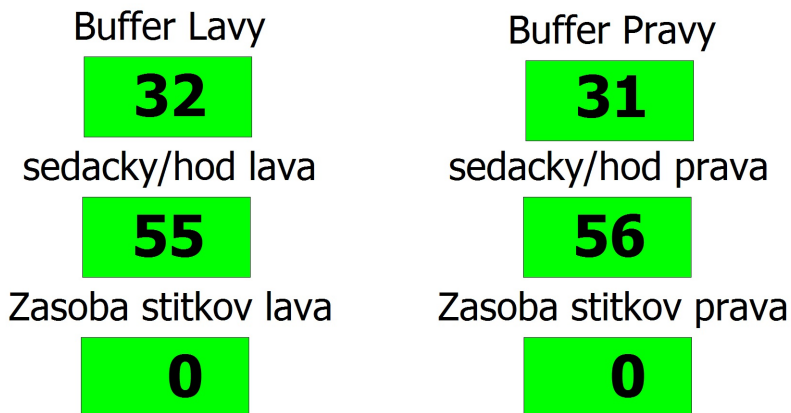
Ďalej sú tu Button (tlačidlá), ktoré slúžia na prepínanie medzi obrazovkami a Date/Time field, ktorý zobrazuje dátum a čas.

```

1 Sub Read_data_from_file()
2 Dim fso, f, ts, field, MyZf
3
4 FileName = "C:\Users\slovak\indexdown.txt"
5
6 ' Catch errors -- Fehler abfangen
7 On Error Resume Next
8
9 ' Create file object -- Objekt erstellen
10 Set fso = CreateObject("Scripting.FileSystemObject")
11 If Err.Number <> 0 Then
12     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
13     Err.Clear
14     Exit Sub
15 End If
16
17 Set f = fso.OpenTextFile(FileName, 1, 0, -2)
18 If Err.Number <> 0 Then
19     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
20     Err.Clear
21     Exit Sub
22 End If
23
24 ' Read values till end of file -- Werte bis zum Dateiende einlesen
25 Do While Not f.AtEndOfStream
26     MyZf = f.ReadLine
27 Loop
28 field = Split(MyZf, ",")
29
30 stitkyL = field(0)
31 stitkyP = field(1)
32
33 ' Tidy up -- Aufräumen
34 f.Close
35 Set f = Nothing
36 Set fso = Nothing
37 ShowSystemAlarm "Readout of the data was successful!"
38
39 End Sub

```

Obr. 2.11: Skript na čítanie dát zo súboru



Obr. 2.12: Prvá obrazovka „Basic“

## 2.7.4 Druhý Screen - PartsBoard

Druhá obrazovka (obr. 2.17) s názvom „PartsBoard“ obsahuje 3 tabuľky, ktoré zobrazujú rannú, poobednú a nočnú zmenu. Tabuľky sú objekt „Graphic view“, sú vytvorené v Exceli spolu so statickým textom, prekopírované do Screenu. Ostatné dynamické prvky sú z väčšej časti I/O fieldy, ktoré zobrazujú hodnoty. Prvý stĺpec „Hod“ je statický text. Druhý stĺpec „Ciel“ je napojený na Tag „HodPlan\_6“, tento Tag je interný a jeho hodnotu počítame vo WinCC prostredníctvom funkcie, ktorá je na obr. 2.13 táto funkcia vypočíta počet kusov sedačiek, ktoré sa majú v danú hodinu vyrobiť vzhľadom na prestávky. Tag „6min“ predstavuje počet výrobných minút v danej hodine a Tag „TaktTime“ predstavuje čas v sekundách za koľko sa vyrobí jedna sedačka. Preto sú výrobné minúty vynásobené 60 aby sme dostali výrobné sekundy, tie podelíme časom výroby jednej sedačky a dostaneme koľko kusov sedačiek máme vyrobiť za danú hodinu. Táto funkcia pokračuje rovnako pre každú hodinu.

```
1 Sub Zostatok()
2
3 SmartTags("HodPlan_6") = Int((SmartTags("6min")*60) / SmartTags("TaktTime"))
4 SmartTags("zostatok_6") = (SmartTags("6min")*60) - Int((SmartTags("6min")*60) / SmartTags("TaktTime")) * SmartTags("TaktTime")
5
6 SmartTags("HodPlan_7") = Int((SmartTags("7min")*60+SmartTags("zostatok_6")) / SmartTags("TaktTime"))
7 SmartTags("zostatok_7") = (SmartTags("7min")*60+SmartTags("zostatok_6")) - Int((SmartTags("7min")*60+
8 SmartTags("zostatok_6")) / SmartTags("TaktTime")) * SmartTags("TaktTime")
9
10 SmartTags("HodPlan_8") = Int((SmartTags("8min")*60+SmartTags("zostatok_7")) / SmartTags("TaktTime"))
11 SmartTags("zostatok_8") = (SmartTags("8min")*60+SmartTags("zostatok_7")) - Int((SmartTags("8min")*60+
12 SmartTags("zostatok_7")) / SmartTags("TaktTime")) * SmartTags("TaktTime")
13
14 SmartTags("HodPlan_9") = Int((SmartTags("9min")*60+SmartTags("zostatok_8")) / SmartTags("TaktTime"))
15 SmartTags("zostatok_9") = (SmartTags("9min")*60+SmartTags("zostatok_8")) - Int((SmartTags("9min")*60+
16 SmartTags("zostatok_8")) / SmartTags("TaktTime")) * SmartTags("TaktTime")
17
18 SmartTags("HodPlan_10") = Int((SmartTags("10min")*60+SmartTags("zostatok_9")) / SmartTags("TaktTime"))
19 SmartTags("zostatok_10") = (SmartTags("10min")*60+SmartTags("zostatok_9")) - Int((SmartTags("10min")*60+
20 SmartTags("zostatok_9")) / SmartTags("TaktTime")) * SmartTags("TaktTime")
21
22 SmartTags("HodPlan_11") = Int((SmartTags("11min")*60+SmartTags("zostatok_10")) / SmartTags("TaktTime"))
23 SmartTags("zostatok_11") = (SmartTags("11min")*60+SmartTags("zostatok_10")) - Int((SmartTags("11min")*60+
24 SmartTags("zostatok_10")) / SmartTags("TaktTime")) * SmartTags("TaktTime")
25
26 SmartTags("HodPlan_12") = Int((SmartTags("12min")*60+SmartTags("zostatok_11")) / SmartTags("TaktTime"))
```

Obr. 2.13: Výsek z funkcie pre výpočet počtu kusov, ktoré potrebujeme vyrobiť a pre výpočet zostatku

Stĺpec „Real“ (obr. 2.17) zobrazuje reálny počet kusov vyrobených sedačiek. Je napojený na Tag „Hod\_6“ z PLC, ktorý čítame z nášho vytvoreného DB. Stĺpec

„Komul Hod“ predstavuje o koľko sme v pluse alebo mínuse od cieľu s reálne vyrobenými sedačkami v danej hodine. Stĺpec „Komul Zmena“ predstavuje o koľko sme v pluse alebo v mínuse priebežne počas celej zmeny. Na ich výpočty opäť slúži funkcia na obr. 2.14. Táto funkcia vyhodnotí hodinu v ktorej sa nachádzame prostredníctvom tagu „Hod.PLC“ a potom aktivuje Tag prislúchajúci predošlej hodine (lebo chceme vedieť či sme boli v pluse alebo mínuse na konci hodiny). Tak ak sme v 7. hodine, potom sa vyhodnotí Tag „H\_Visible\_6“ ako True, čo znamená, že už bude viditeľný a do Tagu „Kom\_H\_6“ sa zapíše výsledok s plusom alebo mínusom za hodinu a do „Kom\_S\_6“ sa zapíše výsledok s plusom alebo mínusom za zmenu. To, že číslo sa bude zapisovať so znamienkom som nastavila vo vlastnostiach I/O field.

```

1 Sub Calculate()
2 If SmartTags("Hod_PLC")= 7 Then
3     SmartTags("H_Visible_6") = True
4 SmartTags("Kom_H_6") = SmartTags("Hod_6") - Int(SmartTags("HodPlan_6"))
5 SmartTags("Kom_S_6") = SmartTags("Kom_H_6")
6 ElseIf SmartTags("Hod_PLC")= 8 Then
7     SmartTags("H_Visible_7") = True
8 SmartTags("Kom_H_7") = SmartTags("Hod_7") - Int(SmartTags("HodPlan_7"))
9 SmartTags("Kom_S_7") = SmartTags("Kom_H_7") + SmartTags("Kom_H_6")
10 ElseIf SmartTags("Hod_PLC")= 9 Then
11     SmartTags("H_Visible_8") = True
12 SmartTags("Kom_H_8") = SmartTags("Hod_8") - Int(SmartTags("HodPlan_8"))
13 SmartTags("Kom_S_8") = SmartTags("Kom_H_8") + SmartTags("Kom_H_7")
14 ElseIf SmartTags("Hod_PLC")= 10 Then
15     SmartTags("H_Visible_9") = True
16 SmartTags("Kom_H_9") = SmartTags("Hod_9") - Int(SmartTags("HodPlan_9"))
17 SmartTags("Kom_S_9") = SmartTags("Kom_H_9") + SmartTags("Kom_H_8")
18 ElseIf SmartTags("Hod_PLC")= 11 Then
19     SmartTags("H_Visible_10") = True
20 SmartTags("Kom_H_10") = SmartTags("Hod_10") - Int(SmartTags("HodPlan_10"))
21 SmartTags("Kom_S_10") = SmartTags("Kom_H_10") + SmartTags("Kom_H_9")
22 ElseIf SmartTags("Hod_PLC")= 12 Then
23     SmartTags("H_Visible_11") = True
24 SmartTags("Kom_H_11") = SmartTags("Hod_11") - Int(SmartTags("HodPlan_11"))
25 SmartTags("Kom_S_11") = SmartTags("Kom_H_11") + SmartTags("Kom_H_10")
26 ElseIf SmartTags("Hod_PLC")= 13 Then
27     SmartTags("H_Visible_12") = True

```

Obr. 2.14: Výsek z funkcie pre výpočet komulatívu za hodinu a priebežne za zmenu

V spodnom riadku (obr. 2.17) sa sčítava počet kusov sedačiek, ktoré máme za cieľ vyrobiť a počet kusov, ktoré sme reálne vyrobili tiež pomocou funkcie (obr. 2.15, obr. 2.16).

```

1 Sub Piece_Per_Shift()
2
3 SmartTags("Ranna_Target") = SmartTags("HodPlan_6") + SmartTags("HodPlan_7") + SmartTags("HodPlan_8") +
4 SmartTags("HodPlan_9") + SmartTags("HodPlan_10") + SmartTags("HodPlan_11") + SmartTags("HodPlan_12") + SmartTags("HodPlan_13")
5 SmartTags("Poobedna_Target") = SmartTags("HodPlan_14") + SmartTags("HodPlan_15") + SmartTags("HodPlan_16")
6 + SmartTags("HodPlan_17") + SmartTags("HodPlan_18") + SmartTags("HodPlan_19") + SmartTags("HodPlan_20") + SmartTags("HodPlan_21")
7 SmartTags("Nocna_Target") = SmartTags("HodPlan_22") + SmartTags("HodPlan_23") + SmartTags("HodPlan_0") +
8 SmartTags("HodPlan_1") + SmartTags("HodPlan_2") + SmartTags("HodPlan_3") + SmartTags("HodPlan_4") + SmartTags("HodPlan_5")
9
10 End Sub

```

Obr. 2.15: Funkcia pre výpočet počtu kusov, ktoré chceme vyrobiť za celú zmenu

```

1 Sub Piece_Make()
2
3 SmartTags("Ranna_Real") = SmartTags("Hod_6") + SmartTags("Hod_7") + SmartTags("Hod_8") +
4 SmartTags("Hod_9") + SmartTags("Hod_10") + SmartTags("Hod_11") + SmartTags("Hod_12") + SmartTags("Hod_13")
5 SmartTags("Poobedna_Real") = SmartTags("Hod_14") + SmartTags("Hod_15") + SmartTags("Hod_16") +
6 SmartTags("Hod_17") + SmartTags("Hod_18") + SmartTags("Hod_19") + SmartTags("Hod_20") + SmartTags("Hod_21")
7 SmartTags("Nocna_Real") = SmartTags("Hod_22") + SmartTags("Hod_23") + SmartTags("Hod_0") +
8 SmartTags("Hod_1") + SmartTags("Hod_2") + SmartTags("Hod_3") + SmartTags("Hod_4") + SmartTags("Hod_5")
9
10 End Sub

```

Obr. 2.16: Funkcia pre výpočet počtu kusov, ktoré reálne vyrobíme za celú zmenu

Opäť sa tu nachádzajú aj tlačidlá na prepínanie medzi obrazovkami a objekt, ktorý zobrazuje dátum a čas.

Hod	Ciel	Real	Komul Hod	Komul Zmena
6:00	45	35	+0	+0
7:00	53	55	+0	+0
8:00	44	45	+0	+0
9:00	53	56	+0	+0
10:00	53	54	+0	+0
11:00	36	34	+0	+0
12:00	53	54	+0	+0
13:00	48	51	+0	+0
		385	384	

Hod	Ciel	Real	Komul Hod	Komul Zmena
14:00	45	50	+5	+5
15:00	53	7		
16:00	44	0		
17:00	53	0		
18:00	53	0		
19:00	36	0		
20:00	53	0		
21:00	48	0		
		385	57	

Hod	Ciel	Real	Komul Hod	Komul Zmena
22:00	45	0		
23:00	53	0		
0:00	44	0		
1:00	53	0		
2:00	53	0		
3:00	36	0		
4:00	53	0		
5:00	45	0		
		385	0	

faurecia

12/14/2015 3:07:54 PM

Setting

Stop

Basic

PartsBoard

Obr. 2.17: Druhá obrazovka „PartsBoard“

### 2.7.5 Tretí Screen - Setting

Tretia obrazovka (obr. 2.18) zobrazuje opäť tri tabuľky, ktoré slúžia na prepočet času. Sú vytvorené v Exceli podobne ako v predchádzajúcom Screene, kde prvé tri stĺpce obsahujú statický text. Ostatné dynamické prvky sú vložené cez objekty. Štvrtý stĺpec predstavuje nevýrobný čas v danej hodine v minútach (prestávky), ktorý zadávame do objektu I/O field ako vstup, napríklad v šiestej hodine naviazaný na Tag „6H“ (v siedmej „7H“ a pod.). Piaty stĺpec predstavuje výrobný čas, ktorý predstavuje počet výrobných minút, teda ak odpočítame nevýrobný čas od 60 minút. Je naviazaný na Tag podľa hodiny (v šiestej hodine „6min“, v siedmej „7min“ a pod.). Posledný stĺpec tabuľky prezentuje výrobný plán v danú hodinu. Je napojený na Tag „HodPlan\_6“ (opäť podľa danej hodiny). A jeho hodnotu rátať podľa funkcie, ktorú som opisovala na obr. 2.13. V tejto funkcii sa ráta aj zostatok, ktorý je zobrazený v oddelenom stĺpci vedľa tabuľky. Zostatok predstavuje sekundy, ktoré sa prenášajú do ďalšej hodiny, lebo sa v danej hodine nestihla dokončiť posledná sedačka a tak sa dorobí v nasledujúcej hodine. V pravej časti



### FIRST SHIFT Ranná zmena

Hour Hodina zo smeny	From Od:	To Do:	Stop time (min) Nevýrobný čas (min)	Worked time (min) Časť prac. čas (min)	Objective Hodnotový plán
1	6:00	7:00	0	52	45
2	7:00	8:00	0	60	53
3	8:00	9:00	0	50	44
4	9:00	10:00	0	60	53
5	10:00	11:00	0	60	53
6	11:00	12:00	0	40	36
7	12:00	13:00	0	60	53
8	13:00	14:00	0	55	48

385

### Lava Linka

Actual Buffer L

Buffer Limit L

Break Time L

Auto Run Sedadlo

Auto Run Operadlo

Read L

Motor L

Sensor

### SECOND SHIFT Poobedná zmena

Hour Hodina zo smeny	From Od:	To Do:	Stop time (min) Nevýrobný čas (min)	Worked time (min) Časť prac. čas (min)	Objective Hodnotový plán
1	14:00	15:00	0	52	45
2	15:00	16:00	0	60	53
3	16:00	17:00	0	50	44
4	17:00	18:00	0	60	53
5	18:00	19:00	0	60	53
6	19:00	20:00	0	40	36
7	20:00	21:00	0	60	53
8	21:00	22:00	0	55	48

385

### Prava Linka

Actual Buffer R

Buffer Limit R

Break Time R

Auto Run Sedadlo

Auto Run Operadlo

Read R

Motor R

Sensor

### THIRD SHIFT Nočná zmena

Hour Hodina zo smeny	From Od:	To Do:	Stop time (min) Nevýrobný čas (min)	Worked time (min) Časť prac. čas (min)	Objective Hodnotový plán
1	22:00	23:00	0	52	45
2	23:00	0:00	0	60	53
3	0:00	1:00	0	50	44
4	1:00	2:00	0	60	53
5	2:00	3:00	0	60	53
6	3:00	4:00	0	40	36
7	4:00	5:00	0	60	53
8	5:00	6:00	0	55	48

385

### Seating

Seat per Day

Cas Taktu : 68.0

Work Time : 1302

Offset : 9

Sedacky/hod plan

Show Num R

Show Num P

Show Num N

Prepočítat

Reset výroby

Stop

Basic

PartBoard

48

### 2.7.6 Štvrtý Screen - StopLine

Štvrtá obrazovka (obr. 2.21) s názvom Stop\_Line, je Screen, ktorý sa aktivuje ak sa vyskytla porucha (zastavená linka pre operadla, pre sedadlá na niektorej stanici). Na obr. 2.21 vidíme stanice linky. Na staniciach P1 až P7 prebieha montáž operadiel a na stanici 1 až 14 montáž sedadiel s operadlami. Je napojená na Tag z PLC, ktorý je spojený s funkciou na obr. 2.19 táto funkcia zabezpečuje prepnutie každej obrazovky na túto obrazovku ak je motor pre linku vypnutý, teda ak BOOL hodnota tohto Tagu je rovná nule. Každá stanica je napojená na svoj Tag, ktorý má BOOL hodnotu a v prípade poruchy je hodnota Tagu True. Pre každý štvorec, ktorý na obr. zobrazuje stanicu je na tento Tag napojené dynamické správanie, ktoré zabezpečí, že v prípade poruchy na určitej stanici sa pozadie štvorca tejto stanice zmení na červené.

```
1 Sub Change_To_Stop_Screen()  
2 If SmartTags("Motor_Run_prava")= True Then  
3   ActivateScreen "Basic", 0  
4 Else  
5   SmartTags("Prestoj_R") = 0  
6   ActivateScreen "Stop_Line", 0  
7 End If  
8 End Sub
```

Obr. 2.19: Funkcia, ktorá aktivuje štvrtú obrazovku v prípade výskytu poruchy

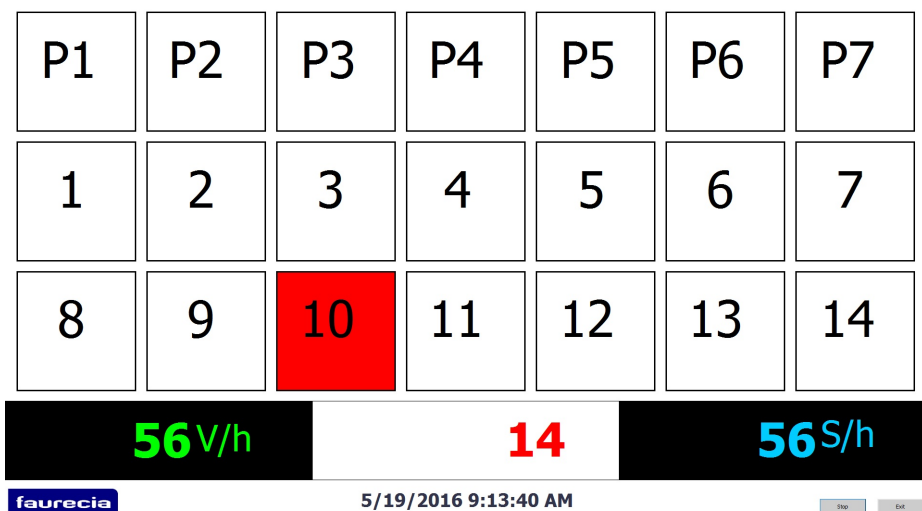
Ďalej sú to I/O field objekty, ktoré zobrazujú rýchlosť výroby sedadiel a operadiel. Táto obrazovka obsahuje aj rátanie času v sekundách, koľko je prevádzka v prestoji (koľko sekúnd nevyrábame). Toto rátanie prestoju je napojené na Tag, ktorého hodnotu rátame pomocou funkcie na obr. 2.20 a táto funkcia je napojená na Tag „ls\_PLC“, ktorý zabezpečuje, že jej vykonanie prebieha každú sekundu.

```

1 Sub Stop_Line_R()
2
3 If SmartTags("Motor_Run_prava")= False Then
4     SmartTags("Prestoj_R") = SmartTags("Prestoj_R") + 1
5 End If
6 End Sub

```

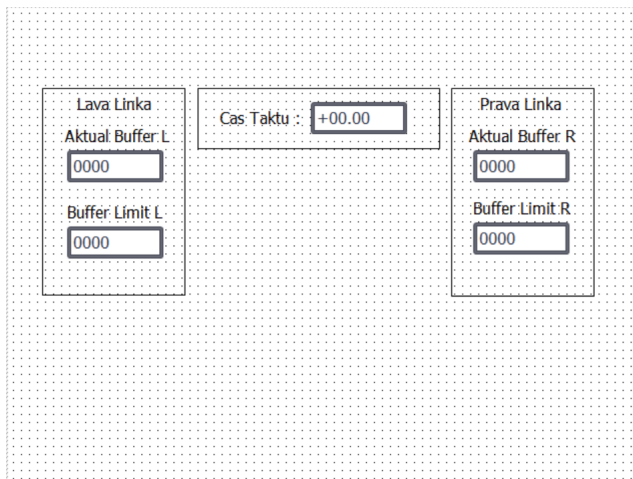
Obr. 2.20: Funkcia pre rátanie sekúnd prestoju



Obr. 2.21: Štvrtá obrazovka StopLine

## 2.8 WinCC na riadenie

Toto WinCC beží na počítači majstra výroby. Na obr. 2.22 vidíme obrazovku pomocou ktorej majster výroby zadáva TaktTime (rýchlosť výroby jednej sedačky), a na základe tohto TaktTime sa prepočíta produkcia pomocou tretieho Screenu Setting (obr. 2.18), ktorý používa funkcie, kde sa ráta s TaktTime (obr.2.13). Ďalej je tu možnosť prepísania počtu sedačiek, ktoré sa nachádzajú v Bufferi. Je to kvôli tomu, že sedačky sa síce môžu vyrobiť ale v prípade zistenia chyby sa z buffera môžu aj odobrať a vtedy majster výroby prepíše toto číslo. Taktiež je tu možné zadávať limity pre Buffre.



Obr. 2.22: Obrazovka pre riadenie produkcie

### 2.8.1 Ďalšie prídavne funkcie

Na prepínanie medzi prvou „Basic“ a druhou „PartsBoard“ obrazovkou slúži funkcia na obr. 2.23. Táto funkcia je napojená na Tag „1m.PLC“, ktorý zabezpečuje, že tieto dve obrazovky sa prepínajú medzi sebou každú minútu.

```

1 Sub Changeing_Screen()
2 |
3 Dim Obj, StrScreenName
4
5 Set Obj = HmiRuntime
6 StrScreenName = Obj.BaseScreenName
7 If StrScreenName = "Basic" Then
8     ActivateScreen "PartsBoard", 0
9 End If
10
11 If StrScreenName = "PartsBoard" Then
12     ActivateScreen "Basic", 0
13 End If
14 End Sub

```

Obr. 2.23: Prepínanie medzi prvou a druhou obrazovkou

Ďalšia prídavná funkcia je, že na začiatku rannej o šiestej hodine sa vyresetujú všetky hodnoty na obrazovke „PartsBoard“, aby sme videli kde sa nachádzame a teda hodnoty sa budú ukazovať postupne podľa toho v akej hodine sa nachádzame. Funkcia, ktorá to zabezpečuje sa nachádza na obr. 2.24.

<pre> 1 Sub Reset_on_Start() 2 If SmartTags("Hod_PLC") = 6 Then 3 SmartTags("Hod_0") = 0 4 SmartTags("Hod_1") = 0 5 SmartTags("Hod_2") = 0 6 SmartTags("Hod_3") = 0 7 SmartTags("Hod_4") = 0 8 SmartTags("Hod_5") = 0 9 SmartTags("Hod_6") = 0 10 SmartTags("Hod_7") = 0 11 SmartTags("Hod_8") = 0 12 SmartTags("Hod_9") = 0 13 SmartTags("Hod_10") = 0 14 SmartTags("Hod_11") = 0 15 SmartTags("Hod_12") = 0 16 SmartTags("Hod_13") = 0 17 SmartTags("Hod_14") = 0 18 SmartTags("Hod_15") = 0 19 SmartTags("Hod_16") = 0 20 SmartTags("Hod_17") = 0 21 SmartTags("Hod_18") = 0 22 SmartTags("Hod_19") = 0 23 SmartTags("Hod_20") = 0 24 SmartTags("Hod_21") = 0 25 SmartTags("Hod_22") = 0 26 SmartTags("Hod_23") = 0 </pre>	<pre> 27 SmartTags("H_Visible_0") = False 28 SmartTags("H_Visible_1") = False 29 SmartTags("H_Visible_2") = False 30 SmartTags("H_Visible_3") = False 31 SmartTags("H_Visible_4") = False 32 SmartTags("H_Visible_5") = False 33 SmartTags("H_Visible_6") = False 34 SmartTags("H_Visible_7") = False 35 SmartTags("H_Visible_8") = False 36 SmartTags("H_Visible_9") = False 37 SmartTags("H_Visible_10") = False 38 SmartTags("H_Visible_11") = False 39 SmartTags("H_Visible_12") = False 40 SmartTags("H_Visible_13") = False 41 SmartTags("H_Visible_14") = False 42 SmartTags("H_Visible_15") = False 43 SmartTags("H_Visible_16") = False 44 SmartTags("H_Visible_17") = False 45 SmartTags("H_Visible_18") = False 46 SmartTags("H_Visible_19") = False 47 SmartTags("H_Visible_20") = False 48 SmartTags("H_Visible_21") = False 49 SmartTags("H_Visible_22") = False 50 SmartTags("H_Visible_23") = False 51 52 End If 53 End Sub </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Obr. 2.24: Reset výroby

## 2.9 Ukladanie dát do Excelu

V TIA Portali je možnosť ukladania dát vo formáte \*.csv. Toto ukladanie je ale obmedzené vzhľadom na štruktúru výsledného dokumentu. Preto som v TIA Portali vytvorila Script, ktorý zabezpečí ukladanie definovaných dát podľa mnou zvolenej štruktúry. Na obr. 2.25 môžeme vidieť tento Script. Celý Script robí to, že na konci nočnej zmeny (začiatku rannej) zapisuje do vytvoreného súboru archive.csv vyrobené počty kusov sedačiek za každú hodinu v danú zmenu. Výsledok ukladania dát je zobrazený na obr. 2.26.

```
1 Sub Write_data_to_file() názov funkcie
2 Dim fso, f, ts vykoná sa vždy na začiatku rannej smeny
3
4 FileName = "C:\Users\slovak\archive.csv" názov súboru, kde sa budú ukladať dáta
5
6 'Catch errors -- Fehler abfangen
7 On Error Resume Next
8
9 'Create object -- Objekt erstellen
10 Set fso = CreateObject("Scripting.FileSystemObject") vytvorenie objektu
11 If Err.Number <> 0 Then
12     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
13     Err.Clear
14     Exit Sub v prípade výskytu chyby do systémových alarmov
15 End If zapiše chybu, číslo chyby a popis chyby
16
17 ' If no file exists, create a new one -- Datei erstellen, wenn keine vorhanden
18 If Not fso.FileExists(FileName) Then ak súbor na ukladanie
19 fso.CreateTextFile FileName dát neexistuje vytvorí ho
20 End If
21
22 Set f = fso.GetFile(FileName)
23 If Err.Number <> 0 Then
24     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
25     Err.Clear
26     Exit Sub
27 End If
28
29 Set ts = f.OpenAsTextStream(8, -2) otvorí súbor na zápis
30 ' mode "8" to append to file
31 If Err.Number <> 0 Then
32     ShowSystemAlarm "Error #" & CStr(Err.Number) & " " & Err.Description
33     Err.Clear
34     Exit Sub
35 End If
36
37 ' Write new set of values into file -- Neue Werte in die Datei schreiben
38 'ts.WriteLine(get_day & "/" & get_month & "/" & get_year) zapiše dáta v nasledovnom formáte
39 ts.WriteLine(Now)
40 ts.WriteLine("Shift" & ";" & "1 Hod" & ";" & "2 Hod" & ";" & "3 Hod" & ";" & "4 Hod" & ";" & "5 Hod" & ";" & "6 Hod" & ";" & "7 Hod" & ";" & "8 Hod")
41 ts.WriteLine("Ranna" & ";" & Hod_6 & ";" & Hod_7 & ";" & Hod_8 & ";" & Hod_9 & ";" & Hod_10 & ";" & Hod_11 & ";" & Hod_12 & ";" & Hod_13)
42 ts.WriteLine("Poobedna" & ";" & Hod_14 & ";" & Hod_15 & ";" & Hod_16 & ";" & Hod_17 & ";" & Hod_18 & ";" & Hod_19 & ";" & Hod_20 & ";" & Hod_21)
43 ts.WriteLine("Vocna" & ";" & Hod_22 & ";" & Hod_23 & ";" & Hod_0 & ";" & Hod_1 & ";" & Hod_2 & ";" & Hod_3 & ";" & Hod_4 & ";" & Hod_5)
44
45 'Tidy up -- Aufräumen
46 f.Close
47 Set ts = Nothing nakoniec nastaví
48 Set f = Nothing premenné na prázdne
49 Set fso = Nothing
50 ShowSystemAlarm "Storage of the data was successful!"
51
52 End Sub
```

Obr. 2.25: Script na zápis dát do Excelu

	A	B	C	D	E	F	G	H	I
1	19.05.2016 14:20								
2	Shift	1 Hod	2 Hod	3 Hod	4 Hod	5 Hod	6 Hod	7 Hod	8 Hod
3	Ranna	4	52	44	57	57	38	52	23
4	Poobedna	0	0	0	0	0	0	0	0
5	Nocna	0	0	0	0	0	0	0	0

Obr. 2.26: Výsledok ukladania dát do Excelu

### 3 Záver

Vďaka tejto diplomovej práci mi bolo umožnené nahliadnuť do časti automobilového priemyslu, kde sa montujú sedačky do áut. Oboznámila som sa s prevádzkou a princípom montáže na linkách. Naučila som sa teóriu ohľadom SCADA systémov, programovateľných logických regulátorov a venovala som sa ich histórii. Naučila som sa vytvoriť projekt v prostredí Simatic Manager, ktorý slúži na vytvorenie programu, ktorý vykonáva PLC. Zaoberala som sa vytvorením hardvérovej konfigurácie, ktorá je nevyhnutná na správne fungovanie PLC. Naučila som sa použiť bitový senzor v praxi na detekciu sedačky. V prostredí Simatic Manager som sa naučila vytvárať dátové bloky a funkcie. Zaoberala som sa rôznymi typmi prepojenia Step 7 s PLC a priemyselnou komunikáciou. Naučila som sa vytvárať vizualizáciu v prostredí TIA Portal WinCC Advanced. Výsledkom mojej práce je vizualizácia produkcie sedačiek v automobilovom priemysle, ktorá zahŕňa rôzne funkcie písané pomocou skriptov, napríklad pre výpočet kusov sedačiek, ktoré sa majú v danú hodinu vyrobiť, rátať čas, koľko sa linka nachádza v stope a podobne. Vo firme sa nachádzajú štyri linky (pre výrobu ľavej prednej sedačky, pravej prednej sedačky, zadných sedačiek a expedičná linka). Táto práca zahŕňa vizualizáciu produkcie pre linku na ktorej sa vyrábajú pravé predné sedačky do áut. Vizualizácie ostatných liniek sú rovnaké až na to, že sú napojené na Tagy z PLC, ktoré prislúchajú danej linke. Ďalej sa tu nachádza ešte jedna vizualizácia, ktorá slúži na zadávanie vstupných údajov pri výrobe ako je rýchlosť výroby jednej sedačky. Ďalej som sa naučila vytvoriť skript, ktorý zapisuje vybrané údaje do Excelu. Ukladaním dát si vieme sprehľadniť dianie v prevádzke a za určitú dobu spraviť štatistiku. Tak vieme zistiť napríklad kedy boli najčastejšie zastavené linky, kedy sme vyrobili najviac sedačiek a podobne. Touto vizualizáciou som prispela k lepšiemu chodu prevádzky. Ľudia vedia presne koľko vyrábajú a koľko by mali vyrábať. Majster tak vie z týchto údajov robiť opatrenia aby bolo vyrobených dostatočný počet kusov sedačiek. Touto prácou som sa oboznámila o praktickom využití programovateľných logických regulátorov a vizualizácii v reálnej prevádzke.



## Literatúra

- [1] Štefan Kožka, Michal Kvasnica: Programovanie PLC SIMATIC 300 (Základná príručka), STU Bratislava, September 2001
  
- [2] KURZ PROGRAMOVÁNÍ PLC SIEMENS SIMATIC S7-300, DÍL 1, 2, 3. [online]. [cit. 2016-2-10]. Dostupné na internete: < [http : //www.foxon.cz/blogs/category/17 – kurzy – programovani – simatic – s7 – 300.html](http://www.foxon.cz/blogs/category/17-kurzy-programovani-simatic-s7-300.html) >
  
- [3] Standard CPUs. [online]. [cit. 2016-8-2]. Dostupné na internete: < [http : //w3.siemens.com/mcms/programmable – logic – controller/en/advanced – controller/s7 – 300/cpu/standard – cpus/pages/default.aspx#SIMATIC\\_20S7\\_300\\_20CPU\\_20319\\_3\\_20PN\\_DP](http://w3.siemens.com/mcms/programmable – logic – controller/en/advanced – controller/s7 – 300/cpu/standard – cpus/pages/default.aspx#SIMATIC_20S7_300_20CPU_20319_3_20PN_DP) >
  
- [4] PLC a ich programovanie - 1. čo je to PLC?. [online]. [cit. 2016-16-3]. Dostupné na internete: < [http : //www.posterus.sk/?p = 6903](http://www.posterus.sk/?p=6903) >
  
- [5] Induktivní senzory. [online]. [cit. 2016-5-11]. Dostupné na internete: < <http://www.ifm.com/products/cz/ds/IM513A.htm> >
  
- [6] Schleppkettenkabel, PROFINET. [online]. [cit. 2016-4-10]. Dostupné na internet: < [https : //www.hagemeyershop.com/Hagemeyer/Kabel – %26 – Leitungen/Datenkabel/Datenkabel/IE – C5DD4UG0100MCSA20 – E%3B – Schleppkettenkabel, – PROFINET, – M12 – IP – 67 – Stift – gerade, – RJ45 – IP – 20, – Cat – 5 – \(ISO – IEC – 11801\) – Cat – 5e – \(TIA – T568 – B\), –, IE – C5DD4UG0100MCSA20 – E/p/5031369?prevPageNumber = 0](https://www.hagemeyershop.com/Hagemeyer/Kabel – %26 – Leitungen/Datenkabel/Datenkabel/IE – C5DD4UG0100MCSA20 – E%3B – Schleppkettenkabel, – PROFINET, – M12 – IP – 67 – Stift – gerade, – RJ45 – IP – 20, – Cat – 5 – (ISO – IEC – 11801) – Cat – 5e – (TIA – T568 – B), –, IE – C5DD4UG0100MCSA20 – E/p/5031369?prevPageNumber = 0) >

- [7] Patch kabel Datacom. [online]. [cit. 2016-4-9]. Dostupné na internet:   
 < [http : //www.gme.cz/patch – kabel – datacom – utp – rj45 – rj45 – 10 – m – seda – p752 – 633](http://www.gme.cz/patch-kabel-datacom-utp-rj45-rj45-10-m-seda-p752-633) >
- [8] Profibus. [online]. [cit. 2016-4-8]. Dostupné na internet: < [https :  
 //cptrade.pl/produkty/siemens/simatic – net/profibus/#prettyPhoto](https://cptrade.pl/produkty/siemens/simatic-net/profibus/#prettyPhoto) >
- [9] Stejnoseměrné napájecí zdroje. [online]. 03/2004. [cit. 2016-2-15]. Dostupné na internet: < [http :  
 //www.siemens.cz/siemjetstorage/files/11238zdroje.pdf](http://www.siemens.cz/siemjetstorage/files/11238zdroje.pdf) >
- [10] Totally Integrated Automation Portal. [online]. [cit. 2016-4-25]. Dostupné na internet: < [http : //stest1.etnetera.cz/ad/current/?ctxnh =  
 2416f2e791&ctxp = home](http://stest1.etnetera.cz/ad/current/?ctxnh=2416f2e791&ctxp=home) >
- [11] Why can you not display an STL program in FBD or in LAD?. [online]. [cit. 2016-3-2]. Dostupné na internet: < [https :  
 //support.industry.siemens.com/cs/document/16726454/why – can – you – not – display – an – stl – program – in – fbd – or – in – lad?dti = 0&lc = en – WW](https://support.industry.siemens.com/cs/document/16726454/why-can-you-not-display-an-stl-program-in-fbd-or-in-lad?dti=0&lc=en-WW) >
- [12] Průmyslová komunikace. [online]. [cit. 2016-5-1]. Dostupné na internet: < [http : //www1.siemens.cz/ad/current/index.php?vw = 0&ctxnh =  
 dd398ced4f&ctxp = home](http://www1.siemens.cz/ad/current/index.php?vw=0&ctxnh=dd398ced4f&ctxp=home) >
- [13] Dostupné na internete: < [http : //ap.urpi.fei.stuba.sk/pkom/html/cvicenie1.htm](http://ap.urpi.fei.stuba.sk/pkom/html/cvicenie1.htm) >

- [14] PLC AUTOMATIZACE. [online]. [cit. 2016-5-10]. Dostupné na internete: < [http : //www.plc – automatizace.cz/knihovna/historie.htm](http://www.plc-automatizace.cz/knihovna/historie.htm) >
- [15] Kábel RS232. [online]. [cit. 2016-5-5]. Dostupné na internete: < [http :  
//www.tvdigitalne.sk/kable/kabel – rs232 – 120/](http://www.tvdigitalne.sk/kable/kabel-rs232-120/) >
- [16] CP 5711. [online]. [cit. 2016-2-4]. Dostupné na internete: < [https :  
//mall.industry.siemens.com/mall/en/WW/Catalog/Products/10044661](https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10044661) >
- [17] Páločný Juraj: Objektovo orientovaný návrh programového vybavenia SCADA (Bakalárska práca), STU Bratislava, 2010