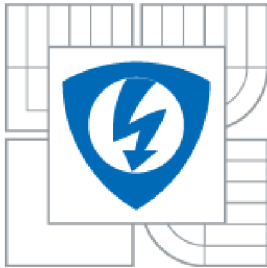




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## VYTVOŘENÍ GENERÁTORU SYMBOLŮ PRO S7- 1200, S7-300/400, S7-1500 DO PROSTŘEDÍ TIA PORTAL V12

GENERATOR OF SYMBOLS FOR S7-1200, S7-300/400, S7-1500 TO TIA PORTAL V12

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

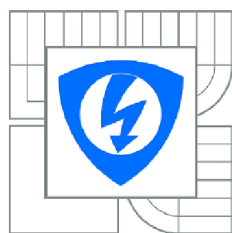
MARTIN HOFEREK

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. JAN PÁSEK, CSc.

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Martin Hoferek  
**Ročník:** 3

**ID:** 146006  
**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Vytvoření generátoru symbolů pro S7-1200, S7-300/400, S7-1500 do prostředí TIA PORTAL V12**

## POKYNY PRO VYPRACOVÁNÍ:

Vytvoření řešení pro zefektivnění přenosu symbolů z elektroprojektu zpracovaného v programu EPLAN P8 do programu PLC SIMATIC S7.

Cíl úkolu:

- Prozkoumat možnosti spolupráce programů EPLAN P8 s prostředím TIA V12 ADVANCED a s notací SCL
- Návrh či úprava nástroje pro generování symbolů z EPLAN P8 do formuláře XLS
- Návrh nástroje pro převedení dat z formuláře XLS do symbolických adres použitelných v programu PLC
- Vytvoření nástrojů, testování pro jednotlivé platformy PLC a vytvoření dokumentace

## DOPORUČENÁ LITERATURA:

- KRÁL, Martin. Excel VBA: Výukový kurz. Brno: Computer Press, 2012. ISBN 978-80-251-2358-4
- LASÁK, Pavel. Stránky o MS Office produktu společnosti Microsoft. [online]. ©2006-2013, aktualizováno 2014-01-01

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 26.5.2014

**Vedoucí práce:** Ing. Jan Pásek, CSc., CSc.

**Konzultanti bakalářské práce:** Ing. Němec Jiří  
Ing. Bezchleba Zbyněk

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## **Abstrakt**

Zvolená bakalářská práce se zabývá návrhem a následně vytvořením nástroje pro generování symbolů do prostředí TIA PORTAL V12. Cílem je zefektivnění přenosu dat mezi elektrotechnickou projekcí a oddělením vývoje software.

## **Klíčová slova**

COMPAS s.r.o., EPLAN P8, PLC SIEMENS, STEP 7, TIA PORTAL V12, Export dat, tabulka symbolických adres

## **Abstract**

The selected bachelor thesis deals with design and following creation tool for generating symbols to the environment TIA PORTAL V12. The aim is to streamline the data transfer between the electrical projections and software development department.

## **Keywords**

COMPAS s.r.o., EPLAN P8, PLC SIEMENS, STEP 7, TIA PORTAL V12, Export data, Table of symbolic addresses

### **Bibliografická citace:**

HOFEREK, M. *Vytvoření generátoru symbolů pro S7-1200, S7-300/400, S7-1500 do prostředí TIA PORTAL V12*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 46s. Vedoucí bakalářské práce byl Ing. Jan Pásek, CSc..

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma **VYTVOŘENÍ GENERÁTORU SYMBOLŮ PRO S7-1200, S7-300/400, S7-1500 DO PROSTŘEDÍ TIA PORTAL V12** jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **22. května 2014**

.....  
podpis autora

## **Poděkování**

Děkuji konzultantům z firmy COMPAS s.r.o. Žďár nad Sázavou pánům Ing. Jiřímu Němcovi a Ing. Zbyňku Bezchlebovi za cenné rady, odbornou pomoc a za to, že se mně snažili vždy vyjít vstříc, kdykoliv jsem něco potřeboval.

Děkuji vedoucímu semestrální práce Ing. Janu Páskovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé semestrální práce.

V Brně dne: **22. května 2014**

.....  
podpis autora

# Obsah

1	Úvod.....	8
2	Eplan Electric P8.....	9
3	Excel VBA .....	11
3.1	Co je to VBA a jak se liší od VB .....	11
3.2	Vytvoření generátoru symbolů.....	11
3.2.1	Projekt elektro .....	12
3.2.2	Konfigurace.....	13
3.2.3	Substituce.....	13
3.2.4	Tabulka symbolů .....	14
3.2.5	Vysvětlení některých pojmů .....	17
3.3	Programování ve VBA.....	18
3.3.1	Proměnné .....	18
3.3.2	Podmínky, cykly .....	19
3.3.3	Důležité funkce a konstrukce .....	20
4	Tvorba aplikace .....	24
4.1	List Projekt elektro – Import souboru .....	24
4.1.1	Import.....	24
4.1.2	Další požadavky .....	25
4.2	List Tabulka symbolů.....	27
4.2.1	PLC Adresa a Datový typ .....	27
4.2.2	Symbolické jméno.....	28
4.2.3	Komentář symbolu .....	31
4.2.4	Manuální výběr .....	34
4.2.5	Další funkce .....	35
4.3	List Konfigurace .....	36
4.4	List Nahradit .....	37
4.5	List Help.....	39
5	Manuál k aplikaci .....	40
5.1	Import souboru z EPLANu .....	40
5.2	Konfigurace manuálního zadávání.....	40
5.3	Tabulka symbolů.....	41
5.4	List Nahradit .....	43
6	Závěr.....	45

# 1 ÚVOD

Téma bakalářská práce je zadáno firmou COMPAS s.r.o., která se zabývá průmyslovou automatizací a dlouhodobě spolupracuje s firmou SIEMENS. Tato práce se zabývá komunikací mezi dvěma stupni vývoje a to elektrotechnickou projekcí a oddělením vývoje software. Elektrotechnická projekce pracuje v programu EPLAN Electric P8. Z výsledného projektu se dá v EPLANu vygenerovat soubor ve formátu XLS (Excel). Tento soubor si vzali programátoři PLC a museli si ho manuálně přetvářet do podoby, kterou potřebovali a kterou pak kopírovali do tabulky symbolů TIA PORTALu (příp. STEP7).

Mým úkolem bude zefektivnit celý postup vytvořením generátoru symbolů. To zrealizuju pomocí maker v prostředí Microsoft Excel 2010. Předtím ještě prozkoumám možnosti exportu dat z EPLANu.

Celý výsledný postup bude tedy vypadat tak, že projekce vygeneruje data z EPLANu pomocí připravené šablony do formátu XLS, na tyto data se spustí makro v Excelu, které převede data do podoby, kterou vyžadují programátoři PLC a ti si je zkopírují do tabulky symbolů TIA PORTALu (příp. STEP7). Pro různé PLC (S7-1200, S7-1500, S7-300/400) bude výsledná podoba tabulky různá.

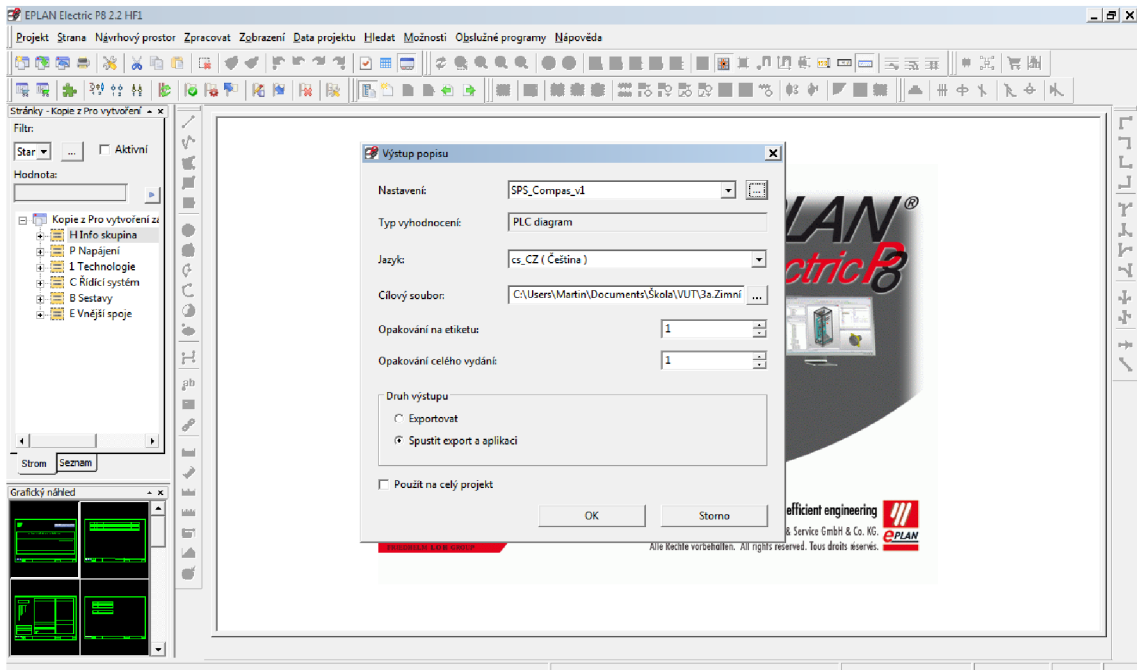
V budoucnu se dá předpokládat, že se v rámci zefektivnění a zrychlení procesu celého vývoje, budou postupně ve firmách zavádět pracovní rámce a platformy jako jsou TIA PORTAL (Siemens) nebo ArchestrA (Wonderware). Ty odstraní rozhraní, které musí při komunikaci mezi elektrotechnickou projekcí a vývojem software překonávat a vytvoří pro ně společný datový prostor. Nyní ovšem pracují obě oddělení v jiných prostředích a touto aplikací by se měl zefektivnit vývoj projektů.



## 2 EPLAN ELECTRIC P8

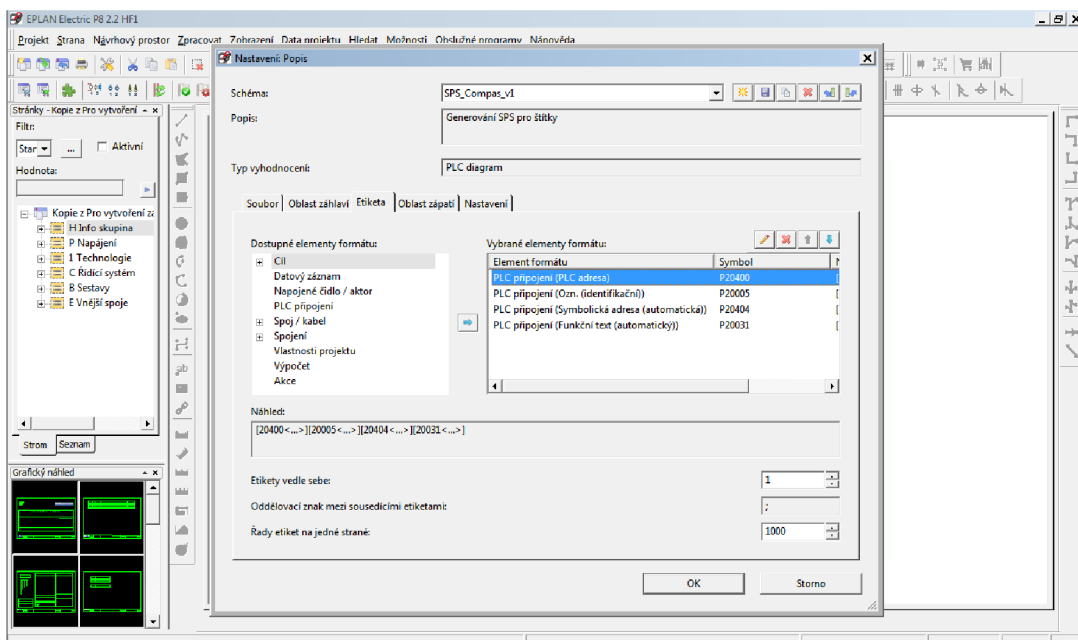
Moje práce se bude EPLANu dotýkat jen okrajově. Jediné co mě na něm bude zajímat, je export dat. Export dat se v EPLANu realizuje pomocí nabídky Obslužné programy -> Vyhodnocení -> Popis. Tím se otevře okno, které je zobrazeno na Obrázek 1.

Obrázek 1: Výstup popisu



Po kliknutí na rozšířené možnosti (tři tečky) se objeví okno, které je na Obrázek 2.

Obrázek 2: Nastavení popisu



Zde se dá buďto manuálně zvolit, které elementy formátu budu chtít do souboru XLS generovat, anebo se dá naimportovat předem vytvořený soubor v jazyce XML, který definuje potřebná data. Právě takový soubor (šablonu) již ve firmě COMPAS s.r.o. používají a to SPS\_Compas\_v1. Ten má vybrané elementy formátu PLC adresu, Identifikační označení, Symbolickou adresu a Funkční text.

Tuto předpřipravenou šablonu jsme se po konzultaci rozhodli nechat ve stejném tvaru, jako byl dosud. Původní představa byla, že by se mohl například rozdělovat funkční text na několik částí již v EPLANu. Po prozkoumání možností EPLANu jsem ovšem zjistil, že takové řešení EPLAN neumožňuje a tak se bude všechna konfigurace provádět až v Excelu.

## **3 EXCEL VBA**

Když se vygeneruje soubor XLS z EPLANu, nainportuje se do souboru Excelu s podporou maker, který bude mým úkolem vytvořit a naprogramovat. V současné době používají ve firmě COMPAS s.r.o. Microsoft Office 2010. Budu tedy makra tvořit a optimalizovat právě pro Microsoft Excel 2010. Makra se píšou v jazyce VBA.

### **3.1 Co je to VBA a jak se liší od VB**

Co je to VBA a jak se liší od VB popisuje Martin Král v knize Excel VBA pro verze 2007 a 2010 takto: „Visual Basic (VB) a Visual Basic for Applications (VBA) mají společný základ. Lze říci, že jádro jazyka Visual Basic 6.0 je součástí instalace Microsoft Office (od verze 2000) a komunikace s ním probíhá přes moduly kódu a uživatelské formuláře, které se od formulářů VB liší svou určitou omezeností, avšak pro většinu aplikací postačí.“ [1]

Pro možnost programování maker je nutno v nastavení Excelu zapnout kartu Vývojář, která je defaultně vypnutá.

### **3.2 Vytvoření generátoru symbolů**

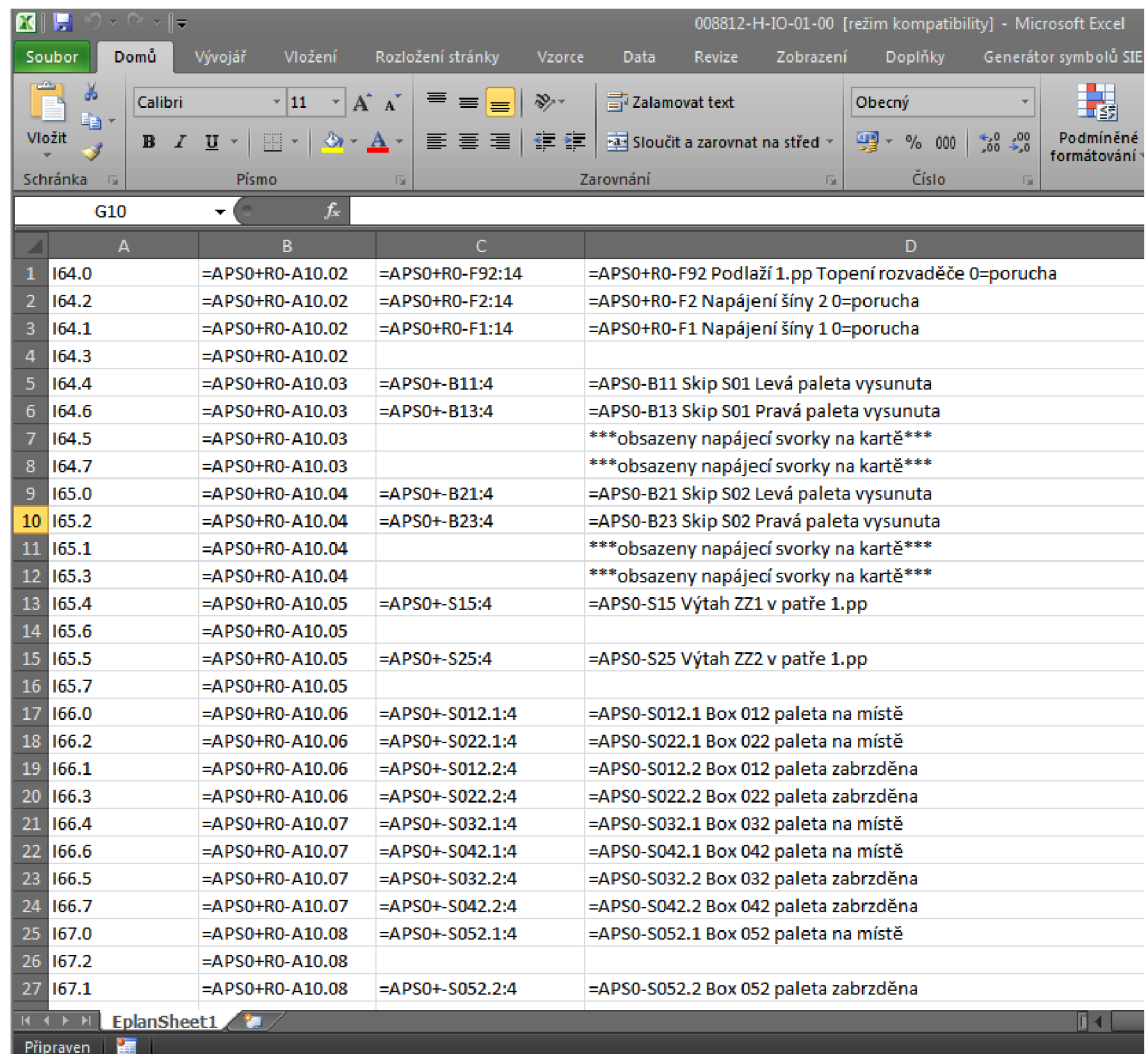
Cílem souboru EXCEL bude tedy vytvořit podkladová data pro tabulku symbolů pro projekt TIA PORTAL (příp. STEP 7). Bude obsahovat následující listy (sheets):

- Projekt elektro
- Konfigurace
- Substituce
- Tabulka symbolů

## 3.2.1 Projekt elektro

Do tohoto listu se bude importovat soubor vygenerovaný EPLANem. Import se bude provádět stisknutím tlačítka, načež se otevře dialogové okno pro výběr souboru, uživatel vybere soubor, který dostal z elektrotechnické projekce a potvrdí. Soubor se naimportuje do stejného listu, tedy „Projekt elektro“. Taký se provede vymazání řádků, kde je prázdná symbolická adresa, což byl další požadavek. V případě že chce naimportovat nový soubor, smaže dosavadní obsah tlačítkem vymazat list. Soubor exportovaný z EPLANu je na Obrázek 3.

Obrázek 3: Soubor exportovaný z EPLANu



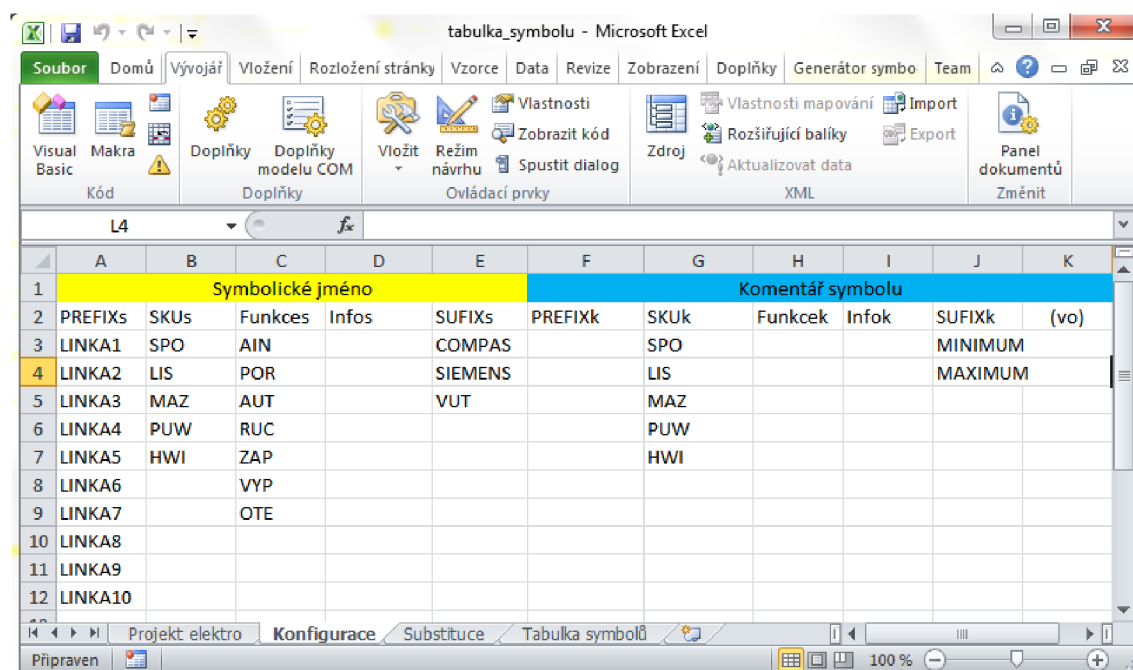
	A	B	C	D
1	I64.0	=APS0+R0-A10.02	=APS0+R0-F92:14	=APS0+R0-F92 Podlaží 1.pp Topení rozvaděče 0=porucha
2	I64.2	=APS0+R0-A10.02	=APS0+R0-F2:14	=APS0+R0-F2 Napájení šíny 2 0=porucha
3	I64.1	=APS0+R0-A10.02	=APS0+R0-F1:14	=APS0+R0-F1 Napájení šíny 1 0=porucha
4	I64.3	=APS0+R0-A10.02		
5	I64.4	=APS0+R0-A10.03	=APS0+-B11:4	=APS0-B11 Skip S01 Levá paleta vysunuta
6	I64.6	=APS0+R0-A10.03	=APS0+-B13:4	=APS0-B13 Skip S01 Pravá paleta vysunuta
7	I64.5	=APS0+R0-A10.03		***obsazeny napájecí svorky na kartě***
8	I64.7	=APS0+R0-A10.03		***obsazeny napájecí svorky na kartě***
9	I65.0	=APS0+R0-A10.04	=APS0+-B21:4	=APS0-B21 Skip S02 Levá paleta vysunuta
10	I65.2	=APS0+R0-A10.04	=APS0+-B23:4	=APS0-B23 Skip S02 Pravá paleta vysunuta
11	I65.1	=APS0+R0-A10.04		***obsazeny napájecí svorky na kartě***
12	I65.3	=APS0+R0-A10.04		***obsazeny napájecí svorky na kartě***
13	I65.4	=APS0+R0-A10.05	=APS0+-S15:4	=APS0-S15 Výtah ZZ1 v patře 1.pp
14	I65.6	=APS0+R0-A10.05		
15	I65.5	=APS0+R0-A10.05	=APS0+-S25:4	=APS0-S25 Výtah ZZ2 v patře 1.pp
16	I65.7	=APS0+R0-A10.05		
17	I66.0	=APS0+R0-A10.06	=APS0+-S012.1:4	=APS0-S012.1 Box 012 paleta na místě
18	I66.2	=APS0+R0-A10.06	=APS0+-S022.1:4	=APS0-S022.1 Box 022 paleta na místě
19	I66.1	=APS0+R0-A10.06	=APS0+-S012.2:4	=APS0-S012.2 Box 012 paleta zabrzděna
20	I66.3	=APS0+R0-A10.06	=APS0+-S022.2:4	=APS0-S022.2 Box 022 paleta zabrzděna
21	I66.4	=APS0+R0-A10.07	=APS0+-S032.1:4	=APS0-S032.1 Box 032 paleta na místě
22	I66.6	=APS0+R0-A10.07	=APS0+-S042.1:4	=APS0-S042.1 Box 042 paleta na místě
23	I66.5	=APS0+R0-A10.07	=APS0+-S032.2:4	=APS0-S032.2 Box 032 paleta zabrzděna
24	I66.7	=APS0+R0-A10.07	=APS0+-S042.2:4	=APS0-S042.2 Box 042 paleta zabrzděna
25	I67.0	=APS0+R0-A10.08	=APS0+-S052.1:4	=APS0-S052.1 Box 052 paleta na místě
26	I67.2	=APS0+R0-A10.08		
27	I67.1	=APS0+R0-A10.08	=APS0+-S052.2:4	=APS0-S052.2 Box 052 paleta zabrzděna

## 3.2.2 Konfigurace

V tomto listu si bude uživatel definovat parametry výsledných řetězců. Zde si uživatel nadefinuje parametry, ze kterých si pak bude vybírat v rolovací nabídce na listu tabulka symbolů. Jednotlivé parametry vysvětlím v kapitole Tabulka symbolů. Možný vzhled je na Obrázek 4.

Dále si zde bude moci uživatel zvolit možnost vyjmutí parametru z některého řetězce. Je potřeba si ještě upřesnit některé možné funkce a varianty co bude a nebude možné.

Obrázek 4: List konfigurace



	Symbolické jméno					Komentář symbolu					
2	PREFIXs	SKUs	Funkces	Infos	SUFIXs	PREFIXk	SKUk	Funkcek	Infok	SUFIXk	(vo)
3	LINKA1	SPO	AIN		COMPAS		SPO			MINIMUM	
4	LINKA2	LIS	POR		SIEMENS		LIS			MAXIMUM	
5	LINKA3	MAZ	AUT		VUT		MAZ				
6	LINKA4	PUW	RUC				PUW				
7	LINKA5	HWI	ZAP				HWI				
8	LINKA6		VYP								
9	LINKA7		OTE								
10	LINKA8										
11	LINKA9										
12	LINKA10										

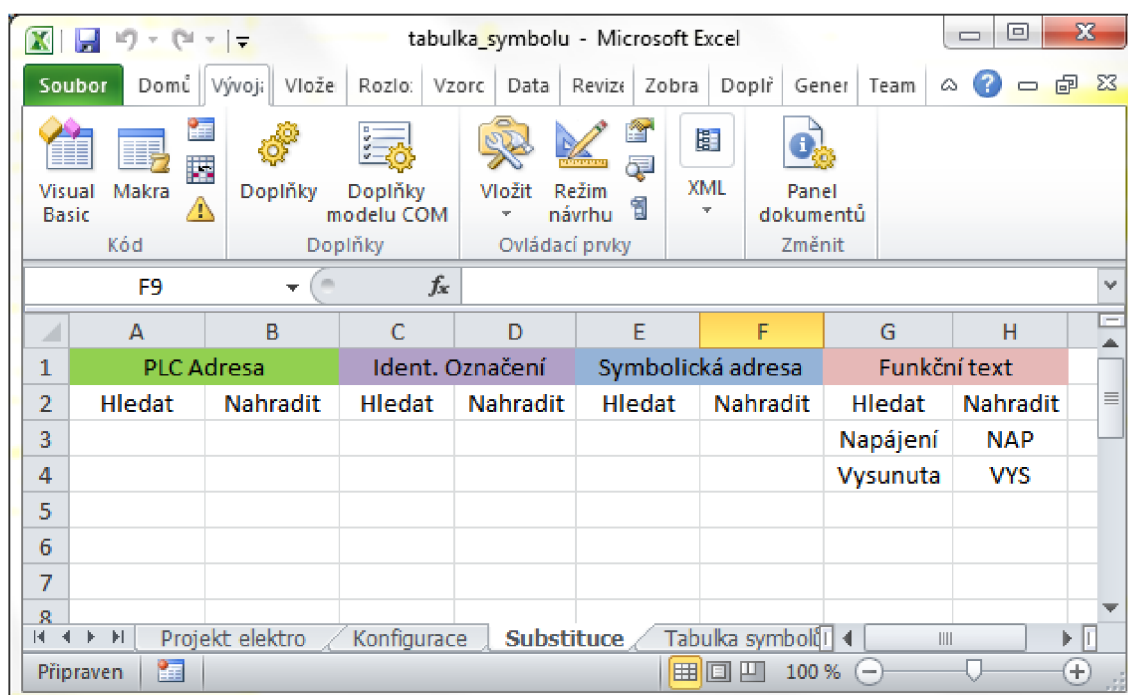
## 3.2.3 Substitute

V tomto listu si uživatel může nadefinovat, které slova bude chtít nahradit. Sloupec Hledat bude mít možnost obsahovat WILD CARD znaky jako například hvězdičku:

- \*a - vezmi znak v poli co je nalevo od a (je-li v poli a)
- a\* - vezmi znak v poli co je napravo od a (je-li v poli a)
- \*a\* - napravo i nalevo
- a?? - všechny znaky napravo od a (je-li v poli a)
- ??a - všechny znaky napravo od a (je-li v poli a)

Tyto možnosti se ještě budou dotvářet a jejich výsledná podoba se změní. Na Obrázek 5 je vidět možný vzhled listu.

Obrázek 5: List Substitute



### 3.2.4 Tabulka symbolů

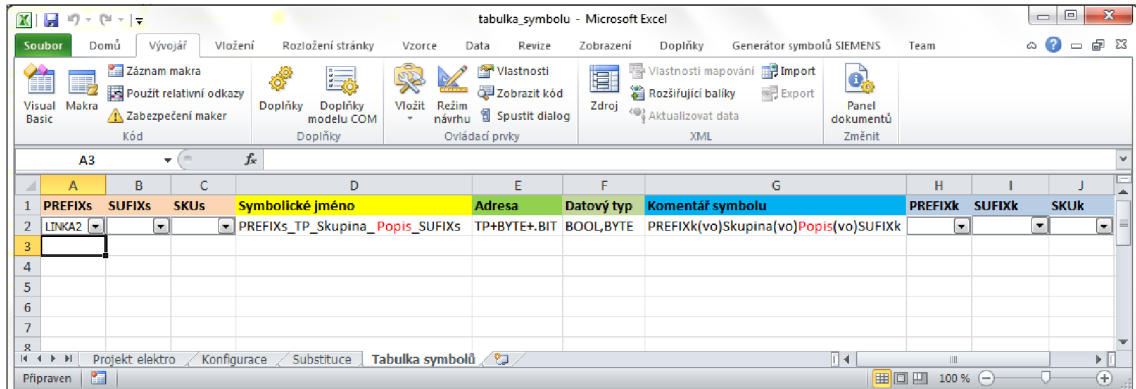
Zde budou umístěna data vygenerovaná pro tabulku symbolů. Tato tabulka bude obsahovat sloupce:

- PREFIXs – volitelný prefix symbolického jména
- SUFIXs – volitelný sufix symbolického jména
- Skupina – specifikace technologické skupiny
- Symbolické jméno
- Adresa
- Datový typ
- Komentář symbolu
- PREFIXk – nepovinná položka např. pro specifikaci tagů do vizualizace
- SUFIXk – nepovinná položka např. pro specifikaci tagů do vizualizace

Možný vzhled listu Tabulky symbolů je na Obrázek 6. Do výsledné tabulky symbolů v TIA PORTAL (příp. STEP 7) se budou kopírovat žlutě vyznačené pojmy, tedy Symbolické jméno, Adresa, Datový typ a Komentář symbolu.

Možný vzhled listu Tabulka symbolů:

Obrázek 6: Sheet tabulka symbolů



### 3.2.4.1 Symbolické jméno

Bude vytvořeno z více řetězců (stringů), které se budou preparovat z položek na listu projekt elektro nebo se manuálně volit. Bude mít následující strukturu:

PREFIX\_TP\_SKU\_Popis\_Info\_SUFFIX

**PREFIX** – Předpona, která může definovat např. úsek ke kterému připoj patří. Volba jen manuálně.

**TP** – Písmena identifikující o jaký datový typ jde např. I,IQ. Generuje se z PLC Adresy.

**SKU** – Skupina, ke které proměnná patří – Generuje se z identifikačního označení na listu Projekt elektro nebo se volí manuálně.

**Popis** – Specifikace zařízení/funkce, generuje se ze Symbolické adresa a určuje zda jde např. o snímač či o jiný prvek.

**Info** – Dodatečná informace upřesňující popis, kterou může uživatel zadat manuálně

**SUFFIX** – Morfém připojený za kořen slova čili koncovka.

Příklad:

„LINKA\_I\_HS\_LAB11\_HladinaHav\_COMPAS“

### 3.2.4.2 Adresa v PLC

Bude se přejímat přímo ze sloupce PLC adresa v listu Projekt elektro. Např. I64.0

### 3.2.4.3 Datový typ

Vychází z adresy PLC, viz Tabulka 1:

Tabulka 1: Datový typ

Adresa	Datový typ
I,Q	BOOL
IB,QB	BYTE
IW,QW	WORD
ID,QD	DWORD

### 3.2.4.4 Komentář symbolu

Bude vytvořeno složením z více stringů a jeho složení bude vypadat takto:

PREFIX\_TP\_SKU\_Popis\_Textzfunkcniho\_SUFIX

**PREFIX** – Předpona, která může definovat např. úsek ke kterému přípoj patří. Volba jen manuálně.

**TP** – Písmena identifikující o jaký datový typ jde např. I,IQ. Generuje se z PLC Adresy.

**SKU** – Skupina, ke které proměnná patří – Generuje se z identifikačního označení na listu Projekt elektro nebo se volí manuálně.

**Popis** – Specifikace zařízení/funkce, generuje se ze Symbolické adresa a určuje zda jde např. o snímač či o jiný prvek.

**Textzfunkcniho** – Text generovaný z konce funkčního textu na listu Projekt elektro.

**SUFIX** – Morfém připojený za kořen slova čili koncovka.

Příklad:

„kom\_pre\_I\_PUW\_B23\_Skip S02 Pravá paleta vysunuta\_MAX“



### **3.2.4.5 Obecné poznámky k tvoření polí**

Manuální výběr pro jednotlivé položky se bude realizovat pomocí comboboxů. Hodnoty těchto comboboxů se budou manuálně zadávat v listu konfigurace. Na listu Tabulka symbolů bude tlačítko, které označí všechna data, která se kopírují do TIA Portalu, tzn. Symbolické jméno, PLC Adresa, Datový typ a Komentář symbolu.

## **3.2.5 Vysvětlení některých pojmů**

V této kapitole bych jenom rád doplnil význam některých zmíněných pojmů.

### **3.2.5.1 Symbolická adresa**

Ke každé PLC adrese náleží symbolická adresa. Obsahuje kódové nebo nějaké zkratkovité označení, které specifikuje např. jeho funkci, ke které kartě náleží atd. Pro programátora je velmi důležité, aby se ve všech proměnných vyznal a zrychlil se tak vývoj software.

### **3.2.5.2 Funkční text**

Má podobnou funkci jako symbolická adresa, ale je ještě konkrétnější. Slovně popisuje, co dané připojení představuje.

### **3.2.5.3 Symbolické jméno**

Zkratkovitý popis označující funkci, jméno a přidělenou skupinu. Může obsahovat doplňující informace jako přípony firmy, která aplikaci vyvíjí atd.

### **3.2.5.4 Komentář symbolu**

Delší popis doplňující symbolické jméno. Obvykle obsahuje slovní popis funkce.

## 3.3 Programování ve VBA

V této kapitole bych rád shrnul některé nejdůležitější příkazy jazyka VBA, které následně použiju při psaní maker.

### 3.3.1 Proměnné

„Při psaní maker je vhodné v úvodu deklarovat proměnné. Sice VBA funguje bez tohoto deklarování, ale zbytečně alokuje pro proměnou větší prostor než je potřeba. Pro malé skripty to nevadí, ale u složitějších programů se bez deklarace neobejdeme. Jinak si zpomalíme výpočet.“ [2]

Ve VBA můžeme proměnným deklarovat tyto platnosti:

- Public – ve všech modulech a procedurách
- Private – pro všechny procedury v daném modulu
- Static – pro jednu proceduru ale po skončení si zachová hodnotu
- Dim – pro jednu proceduru, po skončení se paměť vymaže

Datové typy proměnných ve VBA:

Tabulka 2: Přehled typů proměnných [1]

Datový typ	Rozsah hodnot	Velikost v paměti
Boolean	True nebo False	2 byty
Byte	0-255	1 byte
Currency (Měnový)	+/- 922337203685477.5808	8 bytů
Date (Datum)	1.1 100 až 31.12 9999 0:00:00 23:59:59	8 bytů
Decimal	+/-79228162514264337593543950335 s desetinnou čárkou v libovolném místě	12 bytů
Double	-1.8 až -4.9E324 pro záporná 4.9E324 až 1.8 pro kladná	8 bytů
Integer	+/- 32768	2 byty
Long	+/-2147483648	4 byty
Object	Uložený ukazatel (pointer) na libovolný objekt	4 byty
Single	-3.4E38 až -1.4e-45 pro záporné hodnoty 1.4e-45 až 3.4E38 pro kladné hodnoty	4 byty
String	Může obsahovat až 2 <sup>31</sup> znaků	Různá
User Defined	Může obsahovat jeden nebo více typů	Různá
Variant	Může obsahovat různé hodnoty a objekty	Různá

Deklarace pole se píše:

```
Dim MojePole (0 to 9) As Integer  
Dim MojePole (0 to 9, 0 to 9) As Integer
```

A vepisuje se do něj pak:

```
MojePole(i) = 30
```

## 3.3.2 Podmínky, cykly

Uvedu jen rychlý přehled syntaxí jednotlivých příkazů.

### 3.3.2.1 Podmínka if

```
If podmínka Then  
    Vykonej1  
Elseif podmínka2 Then  
    Vykonej2  
Else  
    Vykonej3  
End If
```

### 3.3.2.2 Sada podmínek Case

```
Select Case Cislo  
Case 1 To 5  
    Vykonej1  
Case 6, 7, 8  
    Vykonej2  
Case Is > 8 And Cislo < 11  
    Vykonej3  
Case Else  
    Vykonej4  
End Select
```

### 3.3.2.3 Konstrukce With, End With

Tato konstrukce je velmi užitečná například ve chvíli, kdy chceme vykonat nějakou sekvenci příkazů např. na listu 1, i když je uživatel přepnutý právě někde jinde. Pak píšeme:

```
With Sheets(List1)
    Vykonej
End With
```

### 3.3.2.4 Cyklus For Next

```
For I = 1 To 10
    Vykonej
Next I
```

### 3.3.2.5 Cyklus Do Loop

```
Do While podminka
    Vykonavej
Loop
```

Nebo

```
Do
    Vykonavej
Loop While podminka
```

Pozn.: Ze všech cyklů můžeme kdykoli vyskočit příkazem Exit For atd.

## 3.3.3 Důležité funkce a konstrukce

Zde bych rád uvedl několik užitečných funkcí a konstrukcí, které budu potřebovat pro splnění všech definovaných úkolů.

### 3.3.3.1 Velká a malá písmena

V jedné z podmínek zadání je napsáno, že má být volitelné, jestli budou mít symbolické jméno a komentář symbolu všechny malá, velká písmena nebo první velké písmeno. Pro tuto operaci má jazyk VBA funkci StrConv. Tato funkce má syntaxi:

```
strOut = StrConv(strIn, Conversion, LocaleID)
```

- strIn – vstupní řetězec, povinné
- Conversion – způsob konverze (vbUpperCase - velká, vbLowerCase - malá, vbProperCase - první velké pro každé slovo), povinné
- LocaleID – národní standart konverze, není-li uveden použije aktuální nastavení

### 3.3.3.2 Dialogové okno

V různých situacích by se mi mohlo hodit, aby při nějaké akci vyskočilo dialogové okno, kde uživatel buď potvrdí nějakou volbu, nebo ho na něco upozorní. To půjde zrealizovat funkcí `msg`. Pokud však budu chtít, aby uživatel zadal např. nějaký text (ale jenom jedna vstupní informace) a abych nemusel vytvářet formulář, vystačí mě metoda `Application.InputBox`.

**Message Box má následující syntaxi:**

`MsgBox`(Prompt, Buttons, Title, Helpfile, Context)

Prompt – Zpráva v dialogu, povinné.

Buttons – Tlačítka, které se zobrazí. Její možnosti jsou v Tabulka 3.

Title – Záhloví dialogu, pokud není uveden tak je tam název aplikace.

Helpfile – Název souboru s nápovědou.

Context – Číslo kontextové nápovědy.

**Tabulka 3: Funkce msg a její parametr Buttons**

Tlačítka	Příkaz
OK	vbOKOnly
OK a Storno	vbOKCancel
Přerušit, Opakovat a Přeskočit	vbAbortRetryIgnore
Ano, Ne a Storno	vbYesNoCancel
Ano, Ne	vbYesNo
Znovu a Storno	vbRetryCancel

Když chci do dialogového okna přidat i ikonu závažnosti zprávy tak k parametru Buttons přidáme se znamínkem + příkazy:

- VbCritical – Kritická zpráva
- VbQuestion – Dotaz
- VbExclamation – Varovná zpráva
- VbInformation – Informační zpráva

Návratovou hodnotu nám potom posílá ve tvaru vbNázevtlačítka např. vbOK nebo vbNo.

### Metoda `InputBox` má syntaxi:

`Application.InputBox`(Prompt, Title, Default, X, Y, HelpFile, HelpContextId, Type)

Prompt – Text zprávy v dialogu

Title – Titulek

Default – Počáteční text zobrazený v „Editu“

X, Y – Pozice okna

HelpFile – Odkaz na soubor s nápovědou

HelpContextId – Číslo tématu nápovědy

Type – Návrátová hodnota Viz Tabulka 4

Tabulka 4: Metoda `InputBox` parametr `Type`

Parametr	Návrátová hodnota
1	Number
2	Text, string
4	Logical value
8	Cell reference
16	Error value
64	Array of values

### 3.3.3.3 Práce s textem

Asi největší částí celé aplikace bude práce s textem. Ať už to bude hledání v řetězci, spojování řetězců, zjišťování jeho délky, porovnávání dvou řetězců, nahrazování jednoho řetězce za druhý nebo jiné. Proto je v této kapitole shrnu, ať se k nim pak můžu při programování vrátit.

Ke zjištění počtu znaků v textu má VBA funkci **Len**, která má následující syntaxi:

```
Delka_retezce = len(retezec)
```

K vyhledávání části řetězce je k dispozici funkce **Instr**:

```
Pozice = Instr(1, „text“, „x“)
```

Když si rozebereme jednotlivé parametry funkce jak jdou za sebou, tak první je pozice, od které bude v řetězci začínat hledat, druhý je ve kterém řetězci bude hledat a třetí je kterou část řetězce má hledat. V příkladu co jsem napsal, by tedy vrátil hodnotu 3. Kdybych dal najít řetězec „ex“ vrátil by hodnotu dva, protože vrací hodnotu začínajícího písmene hledaného řetězce.

Funkce **Left** a **Right** nám vrátí určitý počet znaků zleva nebo zprava. Tedy:

```
Text_zleva = left („text“, 3)
Text_zprava = right („text“, 3)
```

Bude vracet řetězec „ext“ a „tex“.

Pro nalezení znaku nebo části textu a nahrazení jiným znakem nebo částí textu je ve VBA k dispozici funkce **Replace**. Parametry této funkce jsou Textový řetězec, ve kterém budeme nahrazovat znak nebo část textu, který budeme chtít nahradit a čím budeme chtít tento znak nebo část textu nahradit. Funkce má ještě tři nepovinné parametry a to odkud se v textu začne nahrazovat, kolik výskytů budeme nahrazovat a parametr jestli budeme brát zřetel na velká a malá písmena. Syntaxe této funkce je:

```
Replace(Text, co, čím, odkud, kolikrát, porovnání)
Replace („spínač vypnutý“, „vypnutý“, „VYP“)
```

Výsledek bude tedy text „snímač VYP“. Tato funkce se mně bude obzvlášť hodit, protože podobné operace se po mně v aplikaci chtějí. I parametr „kolikrát“ využiju, jelikož je v zadání napsáno, že při hledání bude po nalezení první shody nahrazování ukončeno.

## 4 TVORBA APLIKACE

V této kapitole budu postupně popisovat kroky při vytváření aplikace, překážky, na které narazím a prezentovat konečnou funkčnost jednotlivých částí aplikace.

### 4.1 List Projekt elektro – Import souboru

Jako první krok tvorby aplikace je naprogramovat import souboru, který se vygeneruje z prostředí EPLAN do listu „Projekt elektro“. Požadavky jsou, aby při stisknutí tlačítka vyskočilo dialogové okno pro vybrání souboru, který chci naimportovat a po potvrzení se tento soubor naimportoval. Dalším požadavkem je, aby se smazaly všechny řádky, které mají prázdnou položku „symbolická adresa“. Mohlo by se stát, že jeden soubor z EPLANu bude delší než druhý a tím pádem by zůstávali v listu řádky, které tam nepatří. Toto se ošetří tlačítkem vymazat list, který najde poslední plný řádek a pak smaže celý list. Dalším požadavkem je, aby při rolování listem dolů, byly pořád zobrazeny názvy sloupců.

#### 4.1.1 Import

Aby vyskočilo dialogové okno pro výběr souboru, použiju metodu `Application.GetOpenFilename`. Taky chci, aby měl uživatel možnost zvolit jenom excelovský soubor. To se zadá jako parametr do metody `GetOpenFilename`. Nakonec ještě nastavím, aby se zobrazoval v dialogovém okně nahoře text „Vyberte soubor vygenerovaný z EPLANu“, což se také zadá jako parametr do metody. Konečný tvar pro metodu bude tedy:

```
FName = Application.GetOpenFilename(filefilter:=Filter, Title:=text) kde „Filter“ je řetězec ve tvaru: "Excel Files (*.xls;*.xlsx;*.xlsm;*.xlsb),*.xls;*.xlsx;*.xlsm;*.xlsb" a v textu je již dříve zmíněný text.
```

Pokud uživatel vybere nějaký soubor, vrátí metoda cestu k tomuto souboru, když stiskne tlačítko storno, vrátí hodnotu `FALSE`.

Jestliže mám cestu k souboru, můžu si tento soubor na pozadí otevřít a zkopírovat z něho data, která obsahuje. Nadefinuju si tedy dva workbooky (sešity). Jednomu přiřadím aktivní sešit, tzn. Generátor symbolů a druhému soubor z EPLANu jehož cestu již znám.

To udělám příkazem `Set Eplan = Workbooks.Open(Cestaksouboru)`. Následné kopírování se provede `Eplan.Worksheets(1).Copy after:=TS.Sheets(TS.Sheets.Count)`, což zkopíruje první list na konec listů v aktuálním sešitu. Poté zavřu sešit `Eplan` a v aktivním sešitu překopíruju data z posledního listu do prvního s názvem „Projekt elektro“ a poslední list smažu.



Data v prvním listu ještě upravím tak, aby se smazaly všechny řádky, kde je položka „Symbolická adresa“ prázdná. To udělám cyklem, který od posledního řádku bude porovnávat funkcí `StrComp` buňku C a jestliže je prázdná tak celý řádek smaže. Cyklus musí být od posledního řádku, abych si neposunoval čísla řádků při mazání. Na hodnotu v buňce C se tážu příkazem `Cells(i, "C").Value` a mazání řádku provádím příkazem `Rows(i).Delete`. Během celého makra, a to budu používat často jsem zakázal vykreslování příkazem `Application.ScreenUpdating` nastavením na hodnotu `FALSE`. Díky tomu se všechno děje jenom na pozadí.

### 4.1.2 Další požadavky

Když chce uživatel naimportovat nový list, musí smazat aktuální obsah. Na to slouží tlačítko smazat list. To spustí makro, které nejprve najde poslední řádek, označí celý list a příkazem `Selection.ClearContents` se smaže obsah ve vybrané oblasti.

Další požadavek byl, aby při rolování listem dolů byly pořád viditelné názvy sloupců. To se v Excelu 2010 nastaví v menu Zobrazení, sekce Okno a v nabídce ukotvit příčky se zvolí ukotvit horní řádek.

Konečný vzhled naimportovaných a upravených dat v listu Projekt elektro je na Obrázek 7.

Obrázek 7:List Projekt elektro

Generator symbolů - Microsoft Excel

	A	B	C	D	E	F	G	H
	PLC Adresa	Ident. Označení	Symbolická adresa	Funkční text	Import souboru	Vymazat list		
1								
2	I64.0	=APSO+R0-A10.02	=APSO+R0-F92:14	=APSO+R0-F92 Podlaží 1.pp Topení rozvaděče 0=porucha				
3	I64.2	=APSO+R0-A10.02	=APSO+R0-F2:14	=APSO+R0-F2 Napájení šiny 2 0=porucha				
4	IB120.0	=APSO+R0-A10.02	=APSO+R0-F1:14	=APSO+R0-F1 Napájení šiny 1 0=porucha				
5	QB120.0	=APSO+R0-A10.03	=APSO+ B11:4	=APSO-B11 Skip S01 Levá paleta vysunuta				
6	IW122.0	=APSO+R0-A10.03	=APSO+ B13:4	=APSO-B13 Skip S01 Pravá paleta vysunuta				
7	QW122.0	=APSO+R0-A10.04	=APSO+ B21:4	=APSO-B21 Skip S02 Levá paleta vysunuta				
8	I65.2	=APSO+R0-A10.04	=APSO+ B23:4	=APSO-B23 Skip S02 Pravá paleta vysunuta				
9	ID124.0	=APSO+R0-A10.05	=APSO+ S15:4	=APSO-S15 Výtah Z21 v patře 1.pp				
10	QD124.0	=APSO+R0-A10.05	=APSO+ S25:4	=APSO-S25 Výtah Z22 v patře 1.pp				
11	I66.0	=APSO+R0-A10.06	=APSO+ S012.1:4	=APSO-S012.1 Box 012 paleta na místě				
12	I66.2	=APSO+R0-A10.06	=APSO+ S022.1:4	=APSO-S022.1 Box 022 paleta na místě				
13	I66.1	=APSO+R0-A10.06	=APSO+ S012.2:4	=APSO-S012.2 Box 012 paleta zabrzděna				
14	I66.3	=APSO+R0-A10.06	=APSO+ S022.2:4	=APSO-S022.2 Box 022 paleta zabrzděna				
15	I66.4	=APSO+R0-A10.07	=APSO+ S032.1:4	=APSO-S032.1 Box 032 paleta na místě				
16	I66.6	=APSO+R0-A10.07	=APSO+ S042.1:4	=APSO-S042.1 Box 042 paleta na místě				
17	I66.5	=APSO+R0-A10.07	=APSO+ S032.2:4	=APSO-S032.2 Box 032 paleta zabrzděna				
18	I66.7	=APSO+R0-A10.07	=APSO+ S042.2:4	=APSO-S042.2 Box 042 paleta zabrzděna				
19	I67.0	=APSO+R0-A10.08	=APSO+ S052.1:4	=APSO-S052.1 Box 052 paleta na místě				
20	I67.1	=APSO+R0-A10.08	=APSO+ S052.2:4	=APSO-S052.2 Box 052 paleta zabrzděna				
21	I67.4	=APSO+R0-A10.09	=APSO+ S013.1:4	=APSO-S013.1 Box 013 paleta na místě				
22	I67.6	=APSO+R0-A10.09	=APSO+ S023.1:4	=APSO-S023.1 Box 023 paleta na místě				
23	I67.5	=APSO+R0-A10.09	=APSO+ S013.2:4	=APSO-S013.2 Box 013 paleta zabrzděna				
24	I67.7	=APSO+R0-A10.09	=APSO+ S023.2:4	=APSO-S023.2 Box 023 paleta zabrzděna				
25	I68.0	=APSO+R0-A10.10	=APSO+ S033.1:4	=APSO-S033.1 Box 033 paleta na místě				
26	I68.2	=APSO+R0-A10.10	=APSO+ S053.1:4	=APSO-S053.1 Box 053 paleta na místě				
27	I68.1	=APSO+R0-A10.10	=APSO+ S033.2:4	=APSO-S033.2 Box 033 paleta zabrzděna				
28	I68.3	=APSO+R0-A10.10	=APSO+ S053.2:4	=APSO-S053.2 Box 053 paleta zabrzděna				
29	I68.4	=APSO+R0-A10.11	=APSO+ S014.1:4	=APSO-S014.1 Box 014 paleta na místě				
30	I68.6	=APSO+R0-A10.11	=APSO+ S024.1:4	=APSO-S024.1 Box 024 paleta na místě				
31	I68.5	=APSO+R0-A10.11	=APSO+ S014.2:4	=APSO-S014.2 Box 014 paleta zabrzděna				
32	I68.7	=APSO+R0-A10.11	=APSO+ S024.2:4	=APSO-S024.2 Box 024 paleta zabrzděna				
33	I69.0	=APSO+R0-A10.12	=APSO+ S034.1:4	=APSO-S034.1 Box 034 paleta na místě				
34	I69.2	=APSO+R0-A10.12	=APSO+ S054.1:4	=APSO-S054.1 Box 054 paleta na místě				
35	I69.1	=APSO+R0-A10.12	=APSO+ S034.2:4	=APSO-S034.2 Box 034 paleta zabrzděna				
36	I69.3	=APSO+R0-A10.12	=APSO+ S054.2:4	=APSO-S054.2 Box 054 paleta zabrzděna				
37	I69.4	=APSO+R0-A10.13	=APSO+ S015.1:4	=APSO-S015.1 Box 015 paleta na místě				
38	I69.6	=APSO+R0-A10.13	=APSO+ S025.1:4	=APSO-S025.1 Box 025 paleta na místě				

Projekt elektro Tabulka symbolů Konfigurace Nahrať Help

## 4.2 List Tabulka symbolů

Na tomto listu bude probíhat konečné skládání řetězců a volba jejich parametrů. Obsahuje sloupce, které se budou v konečné fázi kopírovat do TIA portalu. To znamená Symbolické jméno, PLC Adresa, Datový typ a Komentář symbolů. Pro symbolické jméno a komentář symbolů tu jsou sloupce Prefix, Skupina, Suffix a pro symbolické jméno ještě sloupec Info. Tyto sloupce slouží k manuálnímu zadání těchto parametrů. Nyní postupně rozeberu jednotlivé prvky tohoto listu, tzn. algoritmus skládání a naplňování jednotlivých sloupců a funkce tlačítek umístěných na listu.

### 4.2.1 PLC Adresa a Datový typ

Do sloupce PLC adresa se hned při importu souboru nakopírují data z listu Projekt Elektro. Zjištění posledního řádku pro určení oblasti, která se bude kopírovat se děje příkazem `.Cells(Rows.Count, "A").End(xlUp).Row`, ten nám vrátí číslo řádku. Aby sem získal adresu této buňky, spojím číslo převedené na řetězec, s písmenem sloupce:

`Posledni bunka = "A" & CStr(PosledniPlnyRadek)`. Tento proces budu vykonávat u všech sběrů hodnot z různých oblastí. Po naplnění, stále ještě při importu, se začne rozhodovat, co bude ve sloupci datový typ. První fáze je čtení prvního písmene ve sloupci PLC Adresa příkazem `Left(plcadresa, a)`, kde `a` je počet písmen které se budou brát zleva v prvním případě 1. Příkazem `Select case` se provádí třídění. Pro I, Q a D se přiřadí hodnota `BOOL`. Poté se provede druhá fáze třídění. Přečtu tentokrát první dvě písmena a porovná je v `Select case` s hodnotami, přiřadím následovně: `IB, QB BYTE, IW, QW WORD` a `ID, QD DWORD`. Tam kde již druhé písmeno není, zůstane `BOOL` a tam kde je, přiřadí se správná hodnota. Tam, kde je něco jiného, zůstane buňka datový typ prázdná. Příklad v Tabulka 5.

Tabulka 5: Příklad přidělování datového typu

PLC Adresa	Datový typ
I64.0	BOOL
I64.2	BOOL
IB120.0	BYTE
QB120.0	BYTE
IW122.0	WORD
QW122.0	WORD
ID124.0	DWORD
QD124.0	DWORD

## 4.2.2 Symbolické jméno

V této kapitole popíšu způsob a algoritmus skládání symbolického jména. Skládání se provádí podle výše uvedeného schématu: PREFIX\_TP\_SKU\_POPIS\_INFO\_SUFIX. Poskládání slova se provede po stisku tlačítka „Skládej“ ve sloupci Symbolické jméno.

### 4.2.2.1 Prefix symbolického jména

Prefix symbolického jména lze zadat jen manuálně. K tomu slouží sloupec Prefix. Pokud není vyplněn, poskládá se symbolické jméno bez něj. Lze jej napsat manuálně do sloupce A nebo vybrat z comboboxu, který lze vygenerovat tlačítkem „Načti comboboxy pro aktuální řádek“. Položky comboboxu se zadávají na listu „Konfigurace“ do sloupce prefix.

### 4.2.2.2 TP - typ

Získává se z PLC adresy. Jde o počáteční písmena, která určují, zda jde o vstupní bit I, výstupní bit Q, vstupní byte IB, výstupní byte QB, vstupní slovo IW, výstupní slovo QW, vstupní double slovo ID či výstupní double slovo QD. Získám je tím způsobem, že čtu datový typ a je-li BOOL beru jenom první písmeno, jinak beru dvě písmena.

### 4.2.2.3 SKU - skupina

Určuje, do jaké skupiny kontakt patří. Získávám ji z identifikačního označení, kde se nachází mezi znaky = a +. V algoritmu jsem to vyřešil následovně. Příkazem `InStr(1, bunka.Value, "+")` najdu pozici znaku +. Tento příkaz slouží k vyhledání určitého řetězce v jiném řetězci a vrací jeho polohu. Prvním parametrem udávám od kterého písmene má začít prohledávat, druhý parametr říká ve kterém řetězci má hledat a třetí jaký řetězec má hledat. Znak = leží vždy na prvním místě, proto nemusím zjišťovat jeho pozici. Když znám pozice znaků + a = můžu si oddělit část řetězce, kterou potřebuji. Přečtu znaky zleva až po první znak + příkazem `Left(Ident_ozn, pozice_plusu)`. Poté z této části řetězce vezmu zprava všechny znaky kromě rovníčka příkazem `Right(retezec, delka - 1)`. Tím jsem získal skupinu pro symbolické jméno.

Pakliže chce uživatel zadat skupinu manuálně, načte si pro vybraný řádek comboboxy, které mu nabídnou to, co si nadefinoval uživatel v listu konfigurace, a vyberou si nebo prostě manuálně vepíší do sloupce C. Jakmile je vyplněná buňka ve sloupci Skupina tedy sloupci C, tak se Symbolické jméno skládá z tohoto údaje namísto toho získaného z listu projekt elektro.

#### 4.2.2.4 Popis

Určuje, o co se jedná, např. S012.1 určuje, že jde o snímač. Tento údaj se bere ze Symbolické adresy na listu Projekt elektro a nachází se mezi znaky mínus ( - ) a dvojtečka ( : ). Způsob vyjmutí kýžené části textu je podobný jako u skupiny. Tedy najdu pozici znaků dvojtečka a mínus pomocí příkazu `Instr`. Poté vezmu z řetězce zprava počet znaků odpovídající rozdílu délky Symbolické adresy a pozice mínusu. Z této části řetězce se vezme zleva rozdíl pozice dvojtečky a mínusu.

#### 4.2.2.5 Info

Jedná se o dodatečnou informaci upřesňující popis. Lze zadat jen manuálně zase buď pomocí comboboxů nebo vepsáním údaje do kolonky Info.

#### 4.2.2.6 Sufix

Sufix symbolického jména lze zadat jen manuálně a to ve sloupci Sufix. Když se nechá pole prázdné, poskládá se komentář symbolu bez něj.

#### 4.2.2.7 Skládání symbolického jména

Popsal jsem jednotlivé prvky symbolického jména. Teď popíšu algoritmus skládání. Celý proces se spustí po stisknutí tlačítka „Skládej“. Jestliže tedy změním manuálně navolenou hodnotu, je potřeba aktualizovat sloupec symbolické jméno tlačítkem „Skládej“. Jako první se do symbolického jména uloží typ TP. Poté pokud není vyplněna buňka skupina, tak se vezme hodnota získaná z identifikačního označení podle postupu popsaného v kapitole 4.2.2.3. Pokud vyplněná je tak se vezme údaj z ní. Jeden nebo druhý řetězec se připojí s podtržítkem za TP. Za ně se opět s podtržítkem připojí POPIS tedy hodnota získaná ze symbolické adresy podle popisu v kapitole 4.2.2.4. Poté pokud je vyplněná buňka Info připojí se na konec. V závěru se, pokud jsou manuálně navolené, připojí PREFIX a SUFIX. Adresace buněk ze kterých se získávají data manuálně navolená se provádí příkazem `bunka.Offset(, -1).Address`. Tím získám adresu buňky, která je posunutá o jednu buňku vedle na řádku. V případě kladného čísla jde o buňku napravo, v případě záporného o buňku nalevo. Údaje z listu projekt elektro, se kterými pracuju, si ukládám do pole stringů, který se deklaruje například takto `Dim pole_Sku(2 To 5000) As String`. Ukázka oblasti skládání symbolického jména na listu tabulka symbolů je zobrazena na Obrázek 8.

Obrázek 8: Tabulka symbolů - část Symbolické jméno

Generator symbolů - Microsoft Excel

Soubor Domů Vývojář Vložení Rozložení stránky Vzorce Data Revize Zobrazení Doplněk Team

Calibri 11 Zamakovat text Obecný

Vložit Schránka Písmo Zarovnání Číslo Podmíněné formátování Formátovat jako tabulku Styly Vložit Odstranit Buňky

Načti comboboxy pro aktuální řádek				Skládej				Označ data	
Vymaž manuální výběr pro aktuální řádek									
Prefix	Skupina	Info	Suffix	Symbolické jméno				PLC Adresa	Datový typ
				QB_APS0_B11				QB120.0	BYTE
				IW_APS0_B13				IW122.0	WORD
				QW_APS0_B21_ahoj				QW122.0	WORD
LINKA3	MAZ	ZAP	SIEMENS	LINKA3_I_MAZ_B23_ZAP_SIEMENS				165.2	BOOL
				ID_APS0_S15				ID124.0	DWORD
				QD_APS0_S25				QD124.0	DWORD
				I_APS0_S012.1				166.0	BOOL
				I_APS0_S022.1				166.2	BOOL
				I_APS0_S012.2				166.1	BOOL
				I_APS0_S022.2				166.3	BOOL
				I_APS0_S032.1				166.4	BOOL
				I_APS0_S042.1				166.6	BOOL
				I_APS0_S032.2				166.5	BOOL
				I_APS0_S042.2				166.7	BOOL
				I_APS0_S052.1				167.0	BOOL
				I_APS0_S052.2				167.1	BOOL
				I_APS0_S013.1				167.4	BOOL
				I_APS0_S023.1				167.6	BOOL
				I_APS0_S013.2				167.5	BOOL
				I_APS0_S023.2				167.7	BOOL
				I_APS0_S033.1				168.0	BOOL
				I_APS0_S053.1				168.2	BOOL
				I_APS0_S033.2				168.1	BOOL
				I_APS0_S053.2				168.3	BOOL
				I_APS0_S014.1				168.4	BOOL
				I_APS0_S024.1				168.6	BOOL
				I_APS0_S014.2				168.5	BOOL
				I_APS0_S024.2				168.7	BOOL
				I_APS0_S034.1				169.0	BOOL
				I_APS0_S054.1				169.2	BOOL
				I_APS0_S034.2				169.1	BOOL
				I_APS0_S054.2				169.3	BOOL
				I_APS0_S015.1				169.4	BOOL
				I_APS0_S025.1				169.6	BOOL
				I_APS0_S015.2				169.5	BOOL
				I_APS0_S025.2				169.7	BOOL
				I_APS0_S035.1				170.0	BOOL

Projekt elektro Tabulka symbolů Konfigurace Nahradit Help

Připraven

## 4.2.3 Komentář symbolu

V této kapitole popíšu způsob a algoritmus skládání komentáře symbolu. Skládání se provádí podle výše uvedeného schématu:

PREFIX\_TP\_SKU\_POPIIS\_Textzfunckniho\_SUFIX. Poskládání slova se provede po stisku tlačítka „Skládej“ ve sloupci Symbolické jméno.

### 4.2.3.1 Prefix komentáře symbolu

Prefix komentáře symbolu lze zadat jen manuálně a to do sloupce Prefix pro komentář symbolu. Pokud tento údaj uživatel nevyplní, poskládá se komentář symbolu bez prefixu. Je možnost napsat jej manuálně vepsat do sloupce M nebo vybrat z comboboxu, který lze vygenerovat tlačítkem „Načti comboboxy pro aktuální řádek“. Možnosti v comboboxu se zadávají na listu „Konfigurace“ do sloupce prefix pro komentář symbolu.

### 4.2.3.2 TP - typ

Získává se z PLC adresy. Jde o počáteční písmena, která určují, zda jde o vstupní bit I, výstupní bit Q, vstupní byte IB, výstupní byte QB, vstupní slovo IW, výstupní slovo QW, vstupní double slovo ID či výstupní double slovo QD. Získám je tím způsobem, že čtu datový typ a je-li BOOL beru jenom první písmeno, jinak beru dvě písmena.

### 4.2.3.3 SKU - skupina

Údaj, který určuje, do jaké funkční skupiny přípoj patří, např. APS0. Získávám ji z identifikačního označení na listu projekt elektro, kde se nachází mezi znaky = a +. Příkazem `InStr(1, bunka.Value, "+")` najdu pozici znaku +. Tento příkaz slouží k vyhledání určitého řetězce v jiném řetězci a vrátí polohu, na které řetězec začíná. Prvním parametrem se udává od kterého písmene má začít prohledávat, druhý parametr říká ve kterém řetězci má hledat a třetí jaký řetězec má hledat. Znak = leží vždy na prvním místě, proto nemusím zjišťovat jeho pozici. Když znám pozice znaků + a = můžu si oddělit část řetězce, kterou potřebuji. Přečtu znaky zleva až po první znak + příkazem `Left(Ident_ozn, pozice_plusu)`. Poté z této části řetězce vezmu zprava všechny znaky kromě rovnítka příkazem `Right(retezec, delka - 1)`. Když chce uživatel zadat skupinu manuálně, označí kteroukoliv buňku na řádku, kde chce comboboxy načíst, načte si pro vybraný řádek comboboxy tlačítkem „Načti comboboxy pro vybraný řádek“, které mu nabídnou to, co si nadefinoval uživatel v listu konfigurace a vybere si nebo prostě manuálně vepíše do sloupce O. Jakmile je vyplněná buňka ve sloupci Skupina tedy sloupci O tak se komentář symbolu skládá z tohoto údaje namísto toho získaného z listu projekt elektro.

#### 4.2.3.4 Popis

Určuje, o co se jedná, např. S012.1 určuje, že jde o snímač. Tento údaj se bere ze Symbolické adresy na listu Projekt elektro a nachází se mezi znaky mínus ( - ) a dvojtečka ( : ). Způsob vyjmutí kýžené části textu je podobný jako u skupiny. Tedy najdu pozici znaků dvojtečka a mínus pomocí příkazu Instr. Poté vezmu z řetězce zprava počet znaků odpovídající rozdílu délky Symbolické adresy a pozice mínusu. Z této části řetězce se vezme zleva rozdíl pozice dvojtečky a mínusu.

#### 4.2.3.5 Textzfunčniho – Část resp. konec funkčního textu

Popisuje slovně funkci dané adresy. Jedná se o poslední část funkčního textu na listu projekt elektro. Například: „APS0+R0-F92 Podlaží 1.pp Topení rozvaděče 0=porucha“. Poznávacím znakem je první mezera zleva funkčního textu. Zjistím si proto pozici mezery příkazem `InStr(1, funkcni_text.Value, " ")`. Funkce postupuje zleva doprava, takže mi vrátí právě pozici mezery, kterou potřebuji. Z funkčního textu budu brát požadovaný řetězec zprava. Počet znaků, které budu brát zprava bude tedy rozdíl mezi celkovou délkou funkčního textu a pozicí mezery. Požadovanou část funkčního textu tedy získám příkazem `Right(funkcni_text.Value, pozice_1)`.

#### 4.2.3.6 Skládání komentáře symbolů

Skládání komentáře symbolů se spustí po stisknutí tlačítka „Skládej“ ve sloupci komentář symbolu. Pokud jsou prázdné pole pro manuální navolení hodnot, to znamená sloupce M, O, Q, poskládá se komentář symbolu ve tvaru „TP\_SKU\_POPIIS\_Textzfunčniho“. V algoritmu se jako první se do komentáře symbolu uloží typ TP. Poté se připojí skupina získaná z identifikačního označení podle postupu popsaného v kapitole 4.2.3.3. Následně se připojí popis způsobem popsaným v kapitole 4.2.3.4. Jako poslední se připojí část z funkčního textu podle postupu v kapitole 4.2.3.5. Jestliže jsou vyplněny buňky ve sloupcích M, O, Q, tak se připojí prefix na začátek, suffix na konec slova a skupina generovaná z listu projekt elektro bude nahrazena skupina manuálně navolenou. To ovšem až po stisknutí tlačítka skládej ve sloupci komentář symbolu, které aktualizuje celý sloupec. Ukázka oblasti skládání symbolického jména na listu tabulka symbolů je zobrazena na Obrázek 9.



Obrázek 9: Tabulka symbolů - část komentář symbolu

Generator symbolu - Microsoft Excel

	J	K	L	M	N	O	P	Q	R
1	PLC Adresa	Datový typ	Komentář symbolu	Prefix	Skupina	Suffix	Načti výběrové pole pro aktuální řádek Vymaž manuální výběr pro aktuální řádek		
19	I67.0	BOOL	I_APS0_S052.1_Box 052 paleta na místě						
20	I67.1	BOOL	I_APS0_S052.2_Box 052 paleta zabrzděna						
21	I67.4	BOOL	I_APS0_S013.1_Box 013 paleta na místě						
22	I67.6	BOOL	I_APS0_S023.1_Box 023 paleta na místě						
23	I67.5	BOOL	I_APS0_S013.2_Box 013 paleta zabrzděna						
24	I67.7	BOOL	I_APS0_S023.2_Box 023 paleta zabrzděna						
25	I68.0	BOOL	I_APS0_S033.1_Box 033 paleta na místě						
26	I68.2	BOOL	I_APS0_S053.1_Box 053 paleta na místě						
27	I68.1	BOOL	I_APS0_S033.2_Box 033 paleta zabrzděna						
28	I68.3	BOOL	I_APS0_S053.2_Box 053 paleta zabrzděna						
29	I68.4	BOOL	I_APS0_S014.1_Box 014 paleta na místě						
30	I68.6	BOOL	I_APS0_S024.1_Box 024 paleta na místě						
31	I68.5	BOOL	I_APS0_S014.2_Box 014 paleta zabrzděna						
32	I68.7	BOOL	I_APS0_S024.2_Box 024 paleta zabrzděna						
33	I69.0	BOOL	I_APS0_S034.1_Box 034 paleta na místě						
34	I69.2	BOOL	I_APS0_S054.1_Box 054 paleta na místě						
35	I69.1	BOOL	I_APS0_S034.2_Box 034 paleta zabrzděna						
36	I69.3	BOOL	I_APS0_S054.2_Box 054 paleta zabrzděna						
37	I69.4	BOOL	I_APS0_S015.1_Box 015 paleta na místě						
38	I69.6	BOOL	I_APS0_S025.1_Box 025 paleta na místě						
39	I69.5	BOOL	I_APS0_S015.2_Box 015 paleta zabrzděna						
40	I69.7	BOOL	I_APS0_S025.2_Box 025 paleta zabrzděna						
41	I70.0	BOOL	I_APS0_S035.1_Box 035 paleta na místě						
42	I70.2	BOOL	I_APS0_S045.1_Box 045 paleta na místě						
43	I70.1	BOOL	I_APS0_S035.2_Box 035 paleta zabrzděna						
44	I70.3	BOOL	I_APS0_S045.2_Box 045 paleta zabrzděna						
45	I70.4	BOOL	I_APS0_S055.1_Box 055 paleta na místě						
46	I70.5	BOOL	I_APS0_S055.2_Box 055 paleta zabrzděna						
47	I0.0	BOOL	I_APS1_Q1_Hl. vypínač 0=por						
48	I0.2	BOOL	I_APS1_S11_NOT-AUS						
49	I0.1	BOOL	I_APS1_XN_Rezerva						
50	I0.3	BOOL	I_APS1_S11_velin						
51	I0.4	BOOL	I_APS1_Q1_Hl. vypínač 0=por						
52	I0.6	BOOL	I_APS1_S11_NOT-AUS						
53	I0.5	BOOL	I_APS1_XN_Rezerva						
54	I0.7	BOOL	I_APS1_XN_Rezerva						
55	I1.0	BOOL	I_APS1_Q1_Hl. vypínač 0=por						

Projekt elektro Tabulka symbolu Konfigurace Nahraď Help

## 4.2.4 Manuální výběr

Nalevo od symbolického jména a napravo od komentáře symbolů je možnost manuálně naplnit Prefix, Skupinu a Suffix. V případě symbolického jména ještě sloupec Info. Když si chci pro některý řádek vygenerovat comboboxy pro výběr, stisknu tlačítko „Načti comboboxy pro aktuální řádek“. Slova zobrazená v otevírací nabídce comboboxu se definují na kartě konfigurace. Také je možnost vepsat některé slovo přímo do kolonky pro danou část, protože hodnota se nenačítá přímo z comboboxu, ale z vedlejší buňky. Například když chci zadat neobvyklý prefix pro symbolické jméno nebo z jakéhokoli jiného důvodu je připisování na list konfigurace ztrátou času tak se vepíše do sloupce A. Poté stiskne tlačítko „skládej“ a symbolické jméno se poskládá z vepsaného údaje.

### 4.2.4.1 Načtení comboboxů

Pokud se tedy rozhodnu, že chci pro daný řádek načíst rolovací nabídku neboli comboboxy, označím kdekoliv na řádku některou buňku a stisknu tlačítko „Načti comboboxy pro aktuální řádek“. Comboboxy se načtou pro každý sloupec pro symbolické jméno i pro komentář symbolů. Možná někoho napadne, proč se negenerují comboboxy pro všechny řádky automaticky při importu. Je to z toho důvodu, že soubory vygenerované z Eplanu obsahují běžně například 1000 řádků. Při testování generování comboboxů aplikace neodpovídala nebo úplně padala. Po dohodě s panem Bezchlebou jsme zvolili řešení generování comboboxů pro každý řádek zvlášť. Taky ošetřím, aby když už na řádku comboboxy budou, tak aby vyskočila hláška a nepřidaly se další.

Nyní vyberu nejdůležitější prvky kódu pro generování comboboxů. Comboboxy, které se budou generovat na listu, patří do kolekce OLEObjects. Objekt tedy deklaruju jako datový typ OLEObject tzn. `Dim Ole As OLEObject`. Abych ošetřil událost, že se uživatel snaží na řádek, kde již comboboxy jsou, umístit další, použiju příkaz `OnError Resume Next`. Na tento příkaz se pak odvolávám podmínkou `If Err.Number <> 0 Then`, kdy se po splnění této podmínky vykoná vložení comboboxů. To se provádí příkazem `Set Ole = .Parent.OLEObjects.Add`, který vloží do zvolené buňky combobox. Poté objektu ještě nastavím jméno `Ole.name`, zdrojovou oblast, ze které bude combobox brát hodnoty `Ole.ListFillRange` a nastavení výstupní buňky, kde se budou zobrazovat hodnoty zvolené v comboboxu:

```
Ole.LinkedCell = bunka.Offset(, -1).Address(1, 1). Hodnota se
tedy zobrazí nalevo od buňky, kde je combobox. Jestliže není splněna podmínka If
Err.Number <> 0, tak vyskočí hláška message box: „Na tomto řádku již
comboboxy máte“. To se provede příkazem: i = MsgBox("Na tomto řádku
již comboboxy máte.", vbOKOnly + vbInformation,
"Informativní hlášení"). Prvním parametrem udávám řetězec, který se
zobrazí, druhým co se zobrazí za tlačítko a za ikonu u hlášení (v našem případě tedy
```

tlačítko OK a ikonka bílého otazníku v bílém poli) a nakonec se udává hlavička. Pro všechny jednotlivé údaje, tzn. prefix, sufix, skupina atd. se tento algoritmus vykonává stejně.

#### **4.2.4.2 Smazání manuálního výběru**

Pod tlačítkem „Načti comboboxy pro aktuální řádek“ je tlačítko „Vymaž manuální výběr pro aktuální řádek“. To neznamená, že se smažou ze zvoleného řádku comboboxy, ale že se vymažou buňky, ze kterých se hodnoty pro skládání buďto symbolického jména nebo komentáře symbolu berou. Toto tlačítko na rozdíl od tlačítka pro generování comboboxů, je pro symbolické jméno i pro komentář symbolu zvlášť. To znamená, že smaže jen ty údaje, u kterých se nachází.

#### **4.2.5 Další funkce**

Vzhledem k tomu, že list tabulka symbolů je široký a nevejde se na obrazovku, vytvořil sem na tomto listu navigační tlačítka pro posun vpravo, vlevo a zobrazení čtyř sloupců, které se budou kopírovat do TIA portálu, tedy prostřední části. Také jsem naprogramoval tlačítko pro označení právě této prostřední části, která se bude kopírovat do TIA portálu.

##### **4.2.5.1 Označ data**

Sloupce Symbolická adresa, PLC Adresa, Datový typ a Komentář symbolu se budou kopírovat do Eplanu. Pro zjednodušení práce programátora je na listu Tabulka symbolů tlačítko „Označ data“, které označí právě tuto oblast. Zjistí se poslední řádek ve sloupci PLC Adresa, číslo tohoto řádku se sloučí do proměnné „Pos“ s písmenem L (poslední sloupec, který potřebuju) a příkazem `ActiveSheet.Range("I2", Pos).Select` se označí celá oblast.

##### **4.2.5.2 Navigační šipky**

Z výše uvedeného důvodu to znamená, že je list Tabulka symbolů hodně široký, jsem se rozhodl zde umístit navigační šipky. Jsou umístěny ve sloupcích PLC Adresa a Datový typ nahoře. Tyto dva sloupce musí být totiž pořád vidět a to z toho důvodu, že uživatel když tvoří symbolické jméno nebo komentář symbolu, potřebuje vidět pro kterou PLC adresu to tvoří. Pokud tedy stisknu tlačítko doleva „<-“, tak se mi zobrazí oblast pro tvorbu symbolického jména a PLC adresa s datovým typem. Zůstanu na řádku, který jsem měl před stisknutím označen a zobrazí se jako první viditelný řádek nahoře. Zjištění aktuálního řádku se provádí příkazem `ActiveCell.Row`. Přejít na požadovanou oblast se provádí příkazem `Application.GoTo Sheets(2).Range(Pos), True`. Při stisknutí tlačítka doprava „->“ se provede stejný proces jenom na druhou stranu, to znamená pro komentář symbolu. Tlačítko

doprostřed „<->“ zobrazí čtyři sloupce, které se budou kopírovat do TIA portálu. Toto zobrazení může sloužit například ke konečné kontrole vytvořených řetězců.

### 4.3 List Konfigurace

Tento list slouží jako zdrojový soubor pro comboboxy na listu tabulka symbolů. Uživatel si zde nadefinuje, z čeho bude chtít poté vybírat v comboboxech. Údaje na tomto listu zůstávají i po vymazání dat na listu projekt elektro. Konečný vzhled je na Obrázek 10.

Obrázek 10: List Konfigurace konečný vzhled

	A	B	C	D	E	F	G
1	Symbolické jméno				Komentář symbolu		
2	Prefix	Skupina	Info	Suffix	Prefix	Skupina	Suffix
3	LINKA1	SPO	AIN	COMPAS	kom_pre	SPO	MIN
4	LINKA2	LIS	POR	SIEMENS		MAZ	MAX
5	LINKA3	MAZ	AUT	VUT		PUW	
6	LINKA4	PUW	RUC				
7	LINKA5	HWI	ZAP				
8	LINKA6		VYP				
9	LINKA7		OTE				
10	LINKA8						
11	LINKA9						
12	LINKA10						
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							

## 4.4 List Nahradit

Až se poskládají řetězce do požadovaného tvaru, přepne uživatel na kartu nahradit a má možnost najít a nahradit určitou část řetězce jiným řetězcem. V praxi půjde většinou o zkracování slov. Tedy například mám v komentáři symbolu napsáno „I\_APS1\_S11\_NOT-AUS 0=aktivace“ a protože bych to potřeboval zkrátit, nahradím slovo „aktivace“ slovem „akt“. Možná se ptáte, proč by se měla programovat funkce, která je již v Excelu implementována. Požadavek pro tuto funkci byl hlavně kvůli tomu, aby se na jednom místě dalo napsat, co čím chci nahradit, a taky kde, a stiskem jednoho tlačítka se to provedlo.

Pokud mám tedy nadefinováno, co čím budu chtít nahrazovat, stisknu tlačítko „Nahrad' text Hledat textem Nahradit“. Spustí se makro, nahradí se řetězce a přepne se na list Tabulka symbolů, aby mohl provedené změny uživatel vidět, případně zkontrolovat. Důležitá funkce je Instr, která vyhledá zadanou část řetězce a vrátí jeho počáteční pozici. Vzhled listu Nahradit je na Obrázek 11.

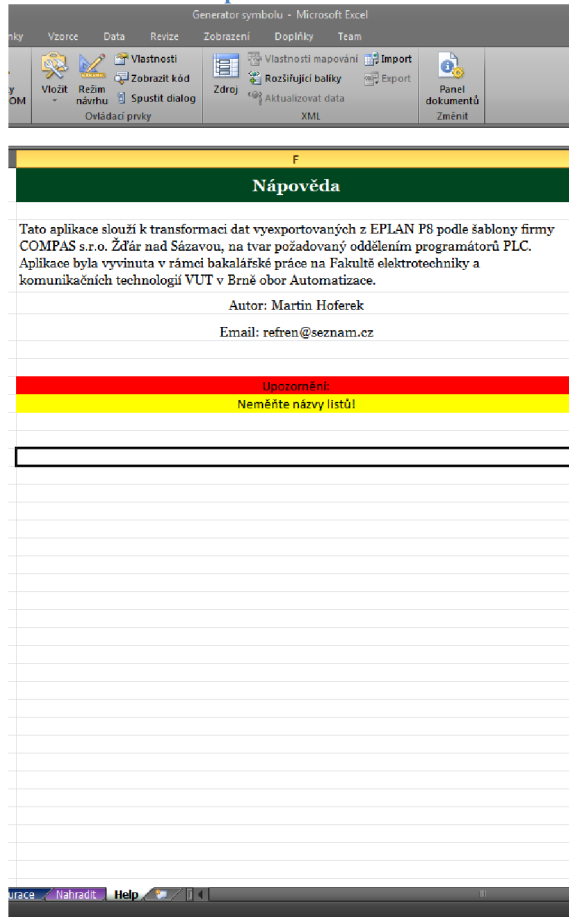
Obrázek 11: List Nahradit - Konečný vzhled

	A	B	C	D
	Nahrad' text Hledat textem Nahradit		Nahrad' text Hledat textem Nahradit	
1	Symbolické jméno		Komentář symbolů	
2	Hledat	Nahradit	Hledat	Nahradit
3	APS0	APS1	porucha	por
4			vysunuta	vysun
5			aktivace	akt
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				

## 4.5 List Help

List Help slouží, jak název napovídá, jako nápověda. Je zde vypsán stručný popis o co jde, kontakt na autora a funkce aplikace. Jeho podoba je na Obrázek 12.

Obrázek 12: List Help



## 5 MANUÁL K APLIKACI

V této kapitole se pokusím o stručný návod k aplikaci. Většina uživatelů, kteří budou potencionálně používat mou aplikaci, nebude potřebovat znát podrobnou funkčnost aplikace, ale budou se s ní chtít seznámit tak, aby ji mohli používat. K tomu bude sloužit právě tato kapitola. Taky se pokusím vše znázornit na obrázcích. Budu se snažit postupovat krok po kroku tak, jak si myslím, že bude konečný uživatel postupovat při práci.

### 5.1 Import souboru z EPLANu

Dříve než začnete s importem souboru, vymažte dosavadní data tlačítkem „Vymazat list“ (Obrázek 14). Pro nahrání zdrojového souboru vygenerovaného EPLANem podle šablony SPS\_Compas\_v1 se stiskne tlačítko Import souboru (Obrázek 13).

Obrázek 13: Tlačítko Import souboru



Obrázek 14: Tlačítko Vymazat list



Data se vám naimportují do listu Projekt elektro, kde si je můžete překontrolovat. Smazaly se řádky, kde nebyla vyplněna symbolická adresa. Dále se vyplnily sloupce PLC Adresa a Datový typ na listu Tabulka symbolů. Nyní se přepněte na list Konfigurace.

### 5.2 Konfigurace manuálního zadávání

Na listu konfigurace si navolte, z čeho si pak chcete na listu Tabulka symbolů vybírat při manuálním zadávání. Možnost manuálního zadávání je pro Prefix, Skupinu, Info a Sufix co se týče symbolického jména a to stejné jenom bez sloupce Info pro komentář symbolů (Obrázek 15). Nyní přepněte na list Tabulka symbolů.



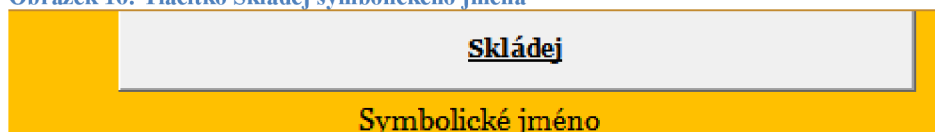
Obrázek 15: Konfigurace

Symbolické jméno				Komentář symbolu		
Prefix	Skupina	Info	Sufix	Prefix	Skupina	Sufix
LINKA1	SPO	AIN	COMPAS	kom_pre	SPO	MIN
LINKA2	LIS	POR	SIEMENS		MAZ	MAX
LINKA3	MAZ	AUT	VUT		PUW	
LINKA4	PUW	RUC				
LINKA5	HWI	ZAP				
LINKA6		VYP				
LINKA7		OTE				
LINKA8						
LINKA9						
LINKA10						

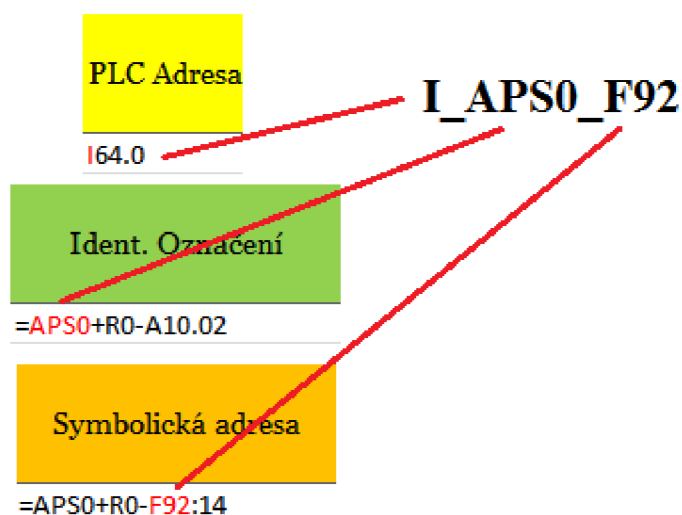
### 5.3 Tabulka symbolů

Nyní přejděme k samotnému skládání a úpravě do téměř konečné podoby. Tlačítkem skládej u symbolického jména (Obrázek 16) se poskládá symbolické jméno ve schématu TP\_SKU\_POPIS to znamená z dat, které se berou ze zdrojového souboru (Obrázek 17).

Obrázek 16: Tlačítko Skládej symbolického jména



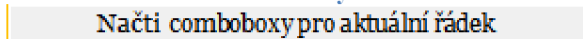
Obrázek 17: Skládání symbolického jména



Tento typ skládání se provede, jestliže jsou pole pro manuální výběr prázdná. Jakmile vyplníte některé z polí, aktualizujte Symbolické jméno tlačítkem „Skládej“. Toto tlačítko aktualizuje všechny hodnoty ve sloupci symbolické jméno.

Pro načtení comboboxů, které umožní výběr z položek nadefinovaných na listu konfigurace, označte jakoukoliv buňku v řádku, ve kterém chcete comboboxy pro výběr načíst a stiskněte tlačítko „Načti comboboxy pro aktuální řádek“ (Obrázek 18). Toto tlačítko načte comboboxy pro celý řádek, tedy i pro Komentář symbolu. Na Obrázek 19 je zobrazeno složení symbolického jména s navolenými hodnotami.

Obrázek 18: Tlačítko comboboxy

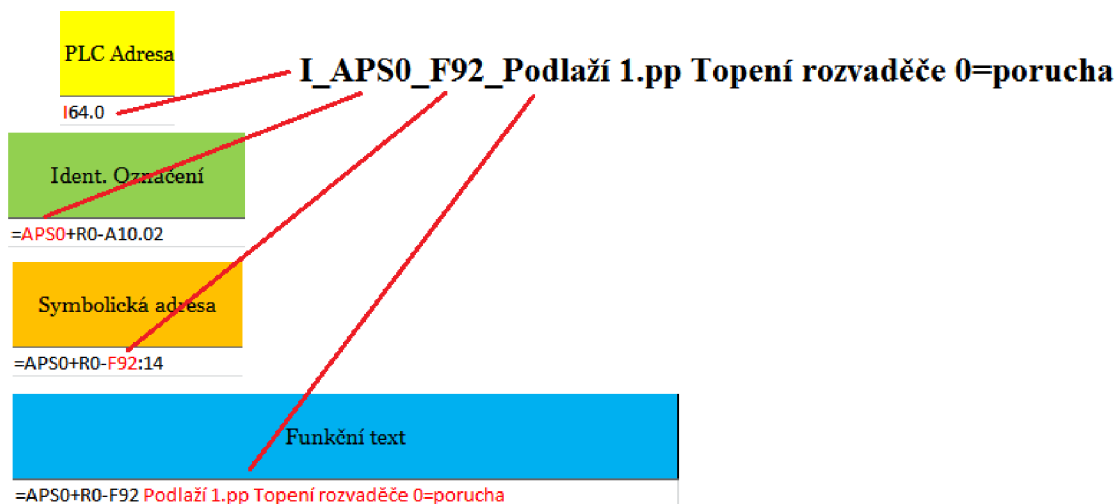


Obrázek 19: Ukázka skládání SJ s navolenými hodnotami

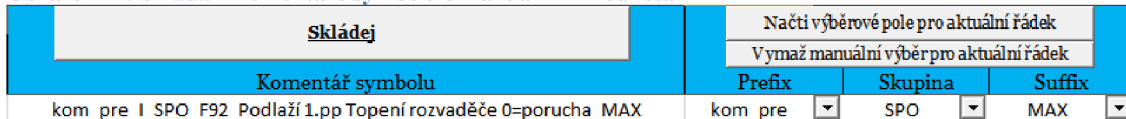


Pokud se uživatel rozhodne, že pro daný řádek manuální hodnoty nechce, tak aby je nemusel mazat jednotlivě, je tady tlačítko „Vymaž manuální výběr pro aktuální řádek“. Stejný princip je i pro komentář symbolu, jen způsob skládání je trochu jiný a taky nelze zvolit manuálně „Info“. Schéma pro tvorbu komentáře symbolu je na Obrázek 20. Ukázka skládání i s manuálně navolenými hodnotami je na Obrázek 21.

Obrázek 20: Skládání komentáře symbolu



Obrázek 21: Skládání komentáře symbolu s manuálními hodnotami



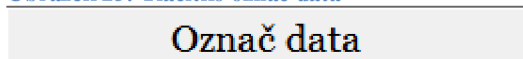
Protože je list tabulka symbolů široký, jsou na něm umístěny navigační tlačítka (Obrázek 22). Tlačítko vlevo zobrazí oblast konstrukce symbolického jména a tlačítko vpravo zobrazí oblast konstrukce komentáře symbolů. U obou zobrazení je pořad viditelná PLC Adresa pro lepší orientaci. Tlačítko uprostřed zobrazí sloupce, které se kopírují do TIA portálu, tzn. Symbolické jméno, PLC Adresu, Datový typ a Komentář symbolu.

Obrázek 22: Navigační šipky



Pro zjednodušení další práce je zde tlačítko „Označ data“ (Obrázek 23), které označí všechny sloupce co se kopírují do TIA Portalu.

Obrázek 23: Tlačítko označ data

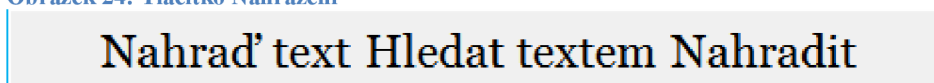


Ještě před kopírováním do TIA Portalu se přepneme na list Nahradit.

## 5.4 List Nahradit

Jestliže jsou výsledné řetězce příliš dlouhé nebo chcete prostě jen nahradit některé slova v symbolickém jménu či komentáři symbolů, tak k tomu slouží list „Nahradit“. Nejčastější způsob bude zřejmě například nahrazení slova za zkratku v komentáři symbolů. Nadefinujte si tedy, co a čím chcete nahradit a stiskněte tlačítko „Nahrad' text Hledat textem Nahradit“ (Obrázek 24). Automaticky vás to přepne na list Tabulka symbolů, kde můžete substituci zkontrolovat. Možný příklad je na Obrázek 25 a výsledek na Obrázek 26 a Obrázek 27.

Obrázek 24: Tlačítko Nahrazení



Obrázek 25: Příklad nahrazování

Symbolické jméno		Komentář symbolů	
Hledat	Nahradit	Hledat	Nahradit
APSO	APS1	porucha	por
		vysunuta	vysun
		aktivace	akt

Obrázek 26: Výsledek substituce A

<b><u>Skládej</u></b>
<b>Symbolické jméno</b>
LINKA3_I_LIS_F92_AUT_SIEMENS
I_APS1_F2
IB_APS1_F1
QB_APS1_B11
IW_APS1_B13
QW_APS1_B21

Obrázek 27: Výsledek substituce B

<b><u>Skládej</u></b>
<b>Komentář symbolu</b>
kom_pre_I_SPO_F92_Podlaží 1.pp Topení rozvaděče 0=por_MAX
I_APS0_F2_Napájení šíny 2 0=por
IB_APS0_F1_Napájení šíny 1 0=por
QB_APS0_B11_Skip S01 Levá paleta vysun
IW_APS0_B13_Skip S01 Pravá paleta vysun
QW_APS0_B21_Skip S02 Levá paleta vysun

## 6 ZÁVĚR

Celá práce byla zadána firmou COMPAS s.r.o. Žďár nad Sázavou a výsledná aplikace se bude používat v reálném provozu. V práci jsem se zabýval zejména návrhem a následně vytvořením této aplikace, která zefektivní přenos dat mezi oddělením elektrotechnické projekce (pracující v prostředí EPLAN P8) a oddělením programování PLC (STEP7 nebo TIA Portál). Aplikace byla vyvíjena v jazyce Visual Basic for Applications, v prostředí Microsoft Excel 2010. Celý proces by měl tedy probíhat tak, že elektrotechnická projekce vygeneruje podle šablony firmy COMPAS, která nese název SPS\_Compas\_v1, soubor (sešit aplikace Microsoft Excel 2010), který předá programátorům PLC. Ti budou uživatelé této aplikace. Soubor si naimportují pomocí jednoho tlačítka. Poté pomocí výše popsaných pomůcek přetvoří data podle své potřeby a následně jsou data připravena pro překopírování do TIA Portálu nebo STEP 7. Na trhu zatím není aplikace, která by tento proces prováděla, bude proto velmi užitečná.

Navržená aplikace zefektivní a usnadní práci, a tedy ušetří čas i peníze. Nasazení v praxi by mělo proběhnout po otestování a posledních úpravách aplikace, které opraví chyby, na které jsem při vývoji nenarazil.

# Literatura

- [1] KRÁL, Martin. *Excel VBA: Výukový kurz*. Brno: Computer Press, 2012. ISBN 978-80-251-2358-4
- [2] LASÁK, Pavel. *Stránky o MS Office produktu společnosti Microsoft*. [online]. ©2006-2013, aktualizováno 2014-01-01 [cit. 2014 01 02]. Dostupné z: <http://office.lasakovi.com/excel/>