

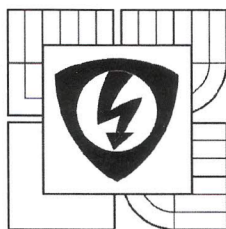
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Michal Burián



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav elektroenergetiky

# Diplomová práce

magisterský navazující studijní obor  
Elektroenergetika

**Student:** Bc. Michal Burián

**Ročník:** 2

**ID:** 134464

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

**Řízení inteligentní elektroinstalace pomocí SCADA systému Reliance**

**POKYNY PRO VYPRACOVÁNÍ:**

1. SCADA systémy – charakteristika, srovnání, využití.
2. Praktická aplikace řízení reálné systémové elektroinstalace Foxtrot s vizualizací.
3. Vzdálené řízení elektroinstalace v reálném čase.

**DOPORUČENÁ LITERATURA:**

podle pokynů vedoucího práce

**Termín zadání:** 8. 2. 2016

**Termín odevzdání:** 20. 5. 2016

**Vedoucí práce:** Ing. Branislav Bátora, Ph.D.

**Konzultanti diplomové práce:**



**doc. Ing. Petr Toman, Ph.D.**

*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být 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.



Bibliografická citace práce:

BURIÁN, M. *Řízení inteligentní elektroinstalace pomocí SCADA systému Reliance*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 66 s. Vedoucí diplomové práce Ing. Branislav Bátora, Ph.D..

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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.

.....



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV ELEKTROENERGETIKY

DEPARTMENT OF ELECTRICAL POWER ENGINEERING

## ŘÍZENÍ INTELIGENTNÍ ELEKTROINSTALACE POMOCÍ SCADA SYSTÉMU RELIANCE

CONTROL OF INTELLIGENT WIRING WITH USED SCADA SYSTEM RELIANCE

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Michal Burián

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Branislav Bátora, Ph.D.

BRNO 2016

## **ABSTRAKT**

Práce se zabývá systémy, které slouží dispečerskému řízení a sběru dat. V teoretické části jsou popsány jednotlivé systémy a jejich srovnání. V praktické části je vytvořen program pro řízení inteligentní elektroinstalace FOXTROT. V programu byla vytvořena vizualizace a vzdálené řízení. Výstupem praktické části je laboratorní návod.

**KLÍČOVÁ SLOVA:** SCADA, Foxtrot, Mosaic, Reliance, Promotic, TIRS.NET

## **ABSTRACT**

This master's thesis deals with systems which are used for supervisory control and data acquisition. The theoretical part concerns with the single systems and their comparison. The practical section describes the designed program for control of smart wiring FOXTROT. The visualization and its remote control were created in this program. The result of practical part is a lab manual.

**KEY WORDS:** SCADA, Foxtrot, Mosaic, Reliance, Promotic, TIRS.NET

## OBSAH

<b>SEZNAM OBRÁZKŮ.....</b>	<b>9</b>
<b>SEZNAM TABULEK .....</b>	<b>11</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK.....</b>	<b>12</b>
<b>1 ÚVOD.....</b>	<b>13</b>
<b>1.1 CÍL PRÁCE .....</b>	<b>13</b>
<b>1.2 SCADA .....</b>	<b>13</b>
<b>1.3 HMI .....</b>	<b>13</b>
<b>1.4 VYUŽITÍ .....</b>	<b>14</b>
<b>2 PŘEHLED A SROVNÁNÍ SCADA/HMI SYSTÉMŮ.....</b>	<b>15</b>
<b>2.1 RELIANCE 4 .....</b>	<b>15</b>
2.1.1 VYUŽITÍ .....	15
2.1.2 STRUKTURA SYSTÉMU .....	15
2.1.3 LICENČNÍ A CENOVÁ POLITIKA .....	18
<b>2.2 PROMOTIC 8.3 .....</b>	<b>21</b>
2.2.1 VYUŽITÍ .....	21
2.2.2 STRUKTURA SYSTÉMU .....	21
2.2.3 LICENČNÍ A CENOVÁ POLITIKA .....	23
<b>2.3 TIRS.NET 6.....</b>	<b>25</b>
2.3.1 VYUŽITÍ .....	25
2.3.2 STRUKTURA SYSTÉMU .....	25
2.3.3 LICENČNÍ A CENOVÁ POLITIKA .....	27
<b>2.4 VÝSLEDNÉ SROVNÁNÍ .....</b>	<b>28</b>
<b>3 PRAKTICKÁ APLIKACE ŘÍZENÍ REÁLNÉ SYSTÉMOVÉ ELEKTROINSTALACE FOXTROT S VIZUALIZACÍ.....</b>	<b>30</b>
<b>3.1 NASTAVENÍ A VYGENEROVÁNÍ SOUBORU S PROMĚNNÝMI.....</b>	<b>31</b>
<b>3.2 VLASTNÍ TVORBA VIZUALIZAČNÍHO PROSTŘEDÍ.....</b>	<b>34</b>
3.2.1 SPRÁVCE STANIC .....	34
3.2.2 SPRÁVCE OBRÁZKŮ.....	36
3.2.3 SPRÁVCE SKRIPTŮ.....	37
3.2.4 VIZUALIZAČNÍ PROSTŘEDÍ.....	39
3.2.5 ŘÍDÍCÍ LOGIKA .....	46
<b>3.3 ŘÍDÍCÍ VIZUALIZAČNÍ PROSTŘEDÍ – POPIS APLIKACE.....</b>	<b>54</b>
<b>4 VZDÁLENÉ ŘÍZENÍ ELEKTROINSTALACE V REÁLNÉM ČASE.....</b>	<b>56</b>
<b>4.1 EXPORT PRO VZDÁLENÉ UŽIVATELE V SYSTÉMU RELIANCE.....</b>	<b>56</b>
<b>4.2 SPUŠTĚNÍ VZDÁLENÝCH KLIENTŮ.....</b>	<b>61</b>
<b>5 ZÁVĚR.....</b>	<b>63</b>
<b>5.1 PŘEHLED A SROVNÁNÍ SCADA/HMI SYSTÉMŮ.....</b>	<b>63</b>

---

<b>5.2 PRAKTICKÁ APLIKACE ŘÍZENÍ REÁLNÉ SYSTÉMOVÉ ELEKTROINSTALACE FOXTROT S VIZUALIZACÍ.....</b>	<b>63</b>
<b>5.3 VZDÁLENÉ ŘÍZENÍ ELEKTROINSTALACE V REÁLNÉM ČASE .....</b>	<b>63</b>
<b>POUŽITÁ LITERATURA .....</b>	<b>65</b>

## SEZNAM OBRÁZKŮ

<i>Obr. 2-1 Koncepce systému TIRS.NET[20]</i> .....	26
<i>Obr. 3-1 Laboratorní panel Foxtrot</i> .....	30
<i>Obr. 3-2 Mosaic - Manažer projektu - Konfigurace HW</i> .....	31
<i>Obr. 3-3 Mosaic - Nastavení V/V – 1</i> .....	32
<i>Obr. 3-4 Mosaic - Nastavení V/V – 2</i> .....	32
<i>Obr. 3-5 Mosaic - Manažer projektu – Exportní soubory</i> .....	33
<i>Obr. 3-6 Reliance Design - Správci</i> .....	34
<i>Obr. 3-7 Reliance Design - Správce stanic – 1</i> .....	35
<i>Obr. 3-8 Reliance Design - Správce stanic – 2</i> .....	36
<i>Obr. 3-9 Reliance Design - Správce stanic – 3</i> .....	36
<i>Obr. 3-10 Reliance Design- Správce obrázků</i> .....	37
<i>Obr. 3-11 Reliance Design- Správce skriptů – 1</i> .....	38
<i>Obr. 3-12 Reliance Design- Správce skriptů – 2</i> .....	38
<i>Obr. 3-13 Reliance Design- Správce skriptů – 3</i> .....	39
<i>Obr. 3-14 Reliance Design- Vkládání komponent – 1</i> .....	40
<i>Obr. 3-15 Reliance Design- Komponenta rám</i> .....	40
<i>Obr. 3-16 Reliance Design- Komponenta displej</i> .....	41
<i>Obr. 3-17 Reliance Design- Komponenta tlačítko – 1</i> .....	41
<i>Obr. 3-18 Reliance Design- Komponenta tlačítko – 2</i> .....	42
<i>Obr. 3-19 Reliance Design – Komponenta obrázek</i> .....	42
<i>Obr. 3-20 Reliance Design – Komponenta aktivní obrázek</i> .....	43
<i>Obr. 3-21 Reliance Design – Žaluzie</i> .....	43
<i>Obr. 3-22 Reliance Design- Komponenta animace</i> .....	44
<i>Obr. 3-23 Reliance Design- Komponenta posuvník</i> .....	44
<i>Obr. 3-24 Reliance Design- Vkládání komponent – 2</i> .....	45
<i>Obr. 3-25 Reliance Design- Komponenta ukazatel – 1</i> .....	45
<i>Obr. 3-26 Reliance Design- Komponenta ukazatel – 2</i> .....	45
<i>Obr. 3-27 Reliance Design- Komponenta ukazatel – 3</i> .....	46
<i>Obr. 3-28 Reliance Design- grafické prvky LED pásků</i> .....	49
<i>Obr. 3-29 Reliance Design- vizualizační prostředí</i> .....	55
<i>Obr. 4-1 Datová komunikace</i> .....	56
<i>Obr. 4-2 Export pro vzdálené uživatele – 1</i> .....	56

---

<i>Obr. 4-3 Export pro vzdálené uživatele – 2.....</i>	<i>57</i>
<i>Obr. 4-4 Export pro vzdálené uživatele – 3.....</i>	<i>57</i>
<i>Obr. 4-5 Export pro vzdálené uživatele – 4.....</i>	<i>58</i>
<i>Obr. 4-6 Export pro vzdálené uživatele – 5.....</i>	<i>58</i>
<i>Obr. 4-7 Export pro vzdálené uživatele – 6.....</i>	<i>59</i>
<i>Obr. 4-8 Export pro vzdálené uživatele – 7.....</i>	<i>59</i>
<i>Obr. 4-9 Export pro vzdálené uživatele – 8.....</i>	<i>60</i>
<i>Obr. 4-10 Export pro vzdálené uživatele – 9.....</i>	<i>60</i>
<i>Obr. 4-11 Webové rozhraní.....</i>	<i>61</i>
<i>Obr. 4-12 Vizualizační prostředí tenkého klienta .....</i>	<i>62</i>



**SEZNAM TABULEK**

<i>Tab. 2-1 Srovnání cen vývojového prostředí Reliance 4 Design verzi Desktop a Enterprise[9]...</i>	19
<i>Tab. 2-2 Srovnání cen běhového prostředí Reliance 4 View, Control, Server, Control Server[9]</i>	19
<i>Tab. 2-3 Srovnání cen balíčků Reliance 4 Combi Package Desktop a Enterprise[9] .....</i>	20
<i>Tab. 2-4 Množstevní ceník tenkých klientů[9].....</i>	20
<i>Tab. 2-5 Srovnání cen běhového prostředí dle počtu proměnných[17].....</i>	24
<i>Tab. 2-6 Množstevní ceník tenkých klientů[17].....</i>	24
<i>Tab. 2-7 Srovnání cen systému TIRS.NET Base v závislosti na počtu datových bodů[24].....</i>	28
<i>Tab. 2-8 Srovnání cen vývojových prostředí jedn. systémů v závislosti na počtu DB[9],[17],[24] .....</i>	29
<i>Tab. 2-9 Srovnání cen běhových prostředí jedn. systémů v závislosti na počtu DB[9], [17], [24] .....</i>	29
<i>Tab. 2-10 Srovnání cen tenkých klientů jedn. systémů v závislosti na počtu klientů[9],[17],[24]</i>	29
<i>Tab. 2-11 Srovnání cen komunikačních driverů jedn. systémů v závisl. na počtu DB[9],[17],[24] .....</i>	29

**SEZNAM SYMBOLŮ A ZKRATEK**

SCADA	Supervisory Control and Data Acquisition
HMI	Human - Machine Interface
MS DOS	Microsoft Disk Operating System
OPC	OLE for Process Control
DDE	Dynamic Data Exchange
VBScript	Visual Basic Scripting Edition
JAVA	Programovací jazyk
PLC	Programmable Logic Controller
BOOL	Datový typ
SW	Software
HW	Hardware
USB	Universal Serial Bus
LPT	Paralelní port
XML	eXtensible Markup Language
DLL	Dynamic Link Library
HTTPS	HyperText Transfer Protocol – Secure
DPH	Daň z přidané hodnoty
RGB	Barevný model červená – zelená - modrá
PC	Osobní počítač
IP	Identifikační číslo síťového rozhraní

# 1 ÚVOD

S postupným rozvojem technické vyspělosti člověka a celé společnosti se rozvíjí také zejména elektrotechnický průmysl a škála výrobků tímto odvětvím produkovaná. Velký rozvoj můžeme pozorovat v oblasti automatizace a inteligentního řízení. Automatizací snižujeme potřebu přítomnosti člověka při vykonávání určité činnosti. V rámci inteligentního řízení máme velké množství výrobců a především jejich výrobků jako programovatelných automatů, snímačů veličin, regulačních prvků, pohonů atd.

Zařízení jednotlivých výrobců používají zpravidla vlastní komunikační protokoly, proto pokud je požadavek na řízení složitější aplikace s různorodými zařízeními je možné k tomu použít nadřazeného SCADA/HMI systému.

## 1.1 Cíl práce

Teoretickým cílem této práce je srovnání jednotlivých SCADA/HMI systémů na základě porovnatelných parametrů. Praktickým cílem je vytvoření vizualizačního prostředí systémové elektroinstalace v systému Reliance 4. A takto vytvořeným vizualizačním prostředím vzdáleně řídit elektroinstalaci v reálném čase.

## 1.2 SCADA

Systémy SCADA (Supervisory Control and Data Acquisition), jak už nám anglický název napovídá, slouží k dispečerskému řízení a sběru dat. Obvyklou aplikací takového systému je monitorování a řízení technologického procesu z centrálního pracoviště, dispečinku.

Ukládání sbíraných dat v těchto systémech je provedeno pomocí databází. Jak už bylo uvedeno výše, výhodou těchto systémů je podpora různorodých komunikačních protokolů, což zvyšuje funkcionalitu celého systému.

## 1.3 HMI

Pojmem HMI (Human - Machine Interface) se označují systémy pro vizualizaci technologických procesů či dat, pracují jako rozhraní mezi zařízením a jeho obsluhou. Vizualizace technologických procesů lze nakonfigurovat tak, aby byly co nejvěrohodnější skutečnosti. Grafické prostředí má velkou výhodu vzhledem k přehlednosti a celkové názornosti technologické situace s čím souvisí případné řešení vzniklých systémových poruch. Tento fakt je poměrně důležitý, jelikož v řadě výrobních procesů může mít prodleva reakce obsluhy na poruchové stavy velmi podstatný vliv na ekonomické následky.

## **1.4 Využití**

Systemy SCADA/HMI se běžně využívají v plynárenství, teplárenství, vodárenství, energetice, dopravě, chemické výrobě, potravinářském průmyslu, těžkém průmyslu, v průmyslových odvětvích využívající výrobní linky, dále v objektech s řízením vytápění a klimatizace, v inteligentních budovách a mnoha dalších odvětvích a aplikacích[1].

## 2 PŘEHLED A SROVNÁNÍ SCADA/HMI SYSTÉMŮ

Softwarů, které pracují na platformě SCADA je značné množství. Mezi nejvýznamnějšího českého zástupce můžeme považovat systém Reliance. Dalšími českými zástupci jsou například systémy Promotic a TIRS.NET. Ze zahraničních můžeme uvést například InTouch, Web Studio, Win CC.

### 2.1 Reliance 4

Systém Reliance je produkt české společnosti GEOVAP, spol. s.r.o. V současné době je na trhu verze pod označením Reliance 4. Avšak historie tohoto systému sahá až do roku 1991, kdy byl na trhu pod označením EP\_DRAW pod platformou MS DOS. Po EP\_DRAW následovaly jednotlivé verze Reliance pod platformou Windows[2].

#### 2.1.1 Využití

V současné době je v tomto systému provozováno více jak 10 000 instalací po celém světě. Nejvzdálenější instalace je řízení systému zavlažování v Austrálii. Nejrozšířenější instalace v ČR je plynárenský dispečink E.On v Českých Budějovicích, kde je instalováno celkem 210 běhových licencí a osmnáct webových klientů.

Mezi zajímavé aplikace můžeme dále uvést vizualizaci zařízení na úpravu vzduchu v Kapli svatého Kříže na hradě Karlštejn, vizualizaci a řízení ropného terminálu Black Sea Terminal v Gruzii a zabezpečení procesu jaderné fúze a operátorů výkonového laseru Ústavu fyziky plazmatu AV ČR[1], [3].

#### 2.1.2 Struktura systému

Systém Reliance je tvořen jednotlivými softwarovými moduly. Zjednodušeně lze rozdělit na vývojové prostředí, běhové prostředí (runtime moduly), tenké klienty, nativní komunikační drivery, OPC a DDE klienta.

##### 2.1.2.1 Vývojové prostředí - Reliance Design

Modul vývojového prostředí, slouží k tvorbě projektu. V modulu se navrhují vizualizační okna s plovoucími nástrojovými okny. Jednotlivé prvky se zde neprogramují, pouze se jedná o vizuální parametrizaci objektů. V případě složitějšího projektu je možnost programovat pomocí skriptů v skriptovacím jazyce VBScript. Modul obsahuje také přehledné nástroje pro správu projektu – správce stanic, správce komunikačních driverů, správce skriptů, správce datových struktur, správce receptur, správce datových tabulek, správce grafů, správce uživatelů, atd. Jazyková podpora vývojového prostředí je v českém a anglickém jazyce.

Modul Design existuje ve dvou verzích a to Desktop a Enterprise. U verze Desktop je možné tvořit aplikace pouze pro jeden PC a libovolný počet stanic, tudíž nelze zde tvořit síťové aplikace

a aplikace pro tenké klienty. U verze Enterprise je možné tvořit aplikace pro více jak jeden PC, tedy síťové aplikace a také aplikace pro tenké klienty[4].

### **2.1.2.2 Běhové prostředí (runtime moduly)**

Jedná se celkem o čtyři moduly, které umožňují běh projektu na úrovni dispečerského řízení, tedy u koncového uživatele. Všechny čtyři moduly mají společné jádro, je tu možnost získávání společných dat pomocí komunikačních portů, dále získávání společných alarmů a jejich následné zpracování. Mezi další vlastnosti patří například proběhnutí skriptů, odesílání textových a emailových zpráv. Je zde rozmanitější jazyková podpora, mimo českého a anglického jazyka také sedm dalších[5].

#### **Reliance View**

Modul View jak už je patrné z názvu slouží pouze k nahlížení a neumožňuje měnit technologické parametry. Proto jeho nejvhodnější použití je u dispečinků, kde obsluha nemá oprávnění zasahovat do dané technologie a pouze sleduje její průběh. Případně tento modul mohou využívat vedoucí pracovníci, kteří mají mít přehled o výrobě a současném stavu technologie, ale nemají oprávnění zasahovat dispečerovi do řízení technologie. Tento modul tedy umožňuje grafické zobrazení dat, alarmů událostí a jejich kvitování. Dále zde lze vytvářet výstupy v podobě tabulek, grafů a sestav. Je zde také možné měnit za běhu jazyk programu nebo projektu, přihlašování a odhlašování uživatelů. Tento modul je možné využít i na dotykových obrazovkách díky dostupné virtuální klávesnici[5].

#### **Reliance Control**

Tento modul je téměř totožný s modulem View. Rozdíl oproti modulu View spočívá v tom, že tento modul umožňuje změnu technologických parametrů. Dovoluje tedy dispečerovi zasahovat do probíhající technologie. Jelikož se jedná o systém především pro dispečerské řízení, pak je tento fakt velmi důležitý. Z tohoto důvodu je modul Control nejvíce rozšířený[5].

#### **Reliance Server**

Jedná se o modul běžící jako služba windows. Nemá vizuální prostředí, tudíž nelze vizualizaci na tomto modulu zobrazit. Slouží pouze jako datový server pro jiné běhové moduly nebo tenké klienty. Je zde možnost vykonávat povely jednotlivých klientů. Součástí je zabudovaný webový server[5].

#### **Reliance Control Server**

Tento Modul spojuje v sobě vlastnosti modulů Control a Server. Oproti modulu Server neběží jako služba windows a umožňuje tedy i vizualizaci dat a technologie. Jedná se o modul s největší užitnou hodnotou[5].

### 2.1.2.3 Tencí klienti

Tuto skupinu zastupují tři klienti. Tito klienti nemají plnou funkčnost oproti klasickým běhovým modulům. Jejich výhodou je, že nám odpadá nutnost jejich instalace. Technologické data tito klienti získávají z modulů Server nebo Control Server prostřednictvím internetu či intranetu. Je zde možnost vizualizace technologických dat, grafů, jak plovoucích, tak historických. Dále je možné zobrazit alarmy, které lze také kvitovat. Nejdůležitější funkcí je možnost ovládání technologie, což je vlastnost modulu Control. Je zde také funkce přihlašování a odhlašování uživatelů. Podpora jazyků je zde stejná jako u běhových modulů[6].

### Reliance Web Client

Tento klient je určený pro vizualizaci pomocí internetového prohlížeče. Je vytvořený na platformě JAVA, z tohoto statusu nám vyplývá, že je nezávislý na operačním systému či prohlížeči[6].

### Reliance Mobile Client

Jedná se o klienta, který má malé zastoupení a je na ústupu z důvodu podpory pouze operačních systémů Windows CE a Windows Mobile[6].

### Reliance Smart Client

Smart Client je stejně jako Web Client určen k vizualizaci pomocí internetového prohlížeče. Je navržen zejména pro využití na tabletech a chytrých telefonech, s čím souvisí, že je uzpůsoben pro dotykové ovládání[6].

### 2.1.2.4 Možnosti komunikace

Systém Reliance je kompatibilní s mnoha komunikačními protokoly. Tento fakt zvyšuje funkcionalitu celého systému, jelikož je použitelný pro větší počet zařízení různých výrobců. Komunikace systému Reliance je možná přes nativní komunikační drivery, OPC servery a webové služby.

### Nativní komunikační drivery

Většina driverů slouží ke spojení s PLC nebo telemetrickými stanicemi. Mezi komunikační drivery spadá také SMS driver, který slouží ke komunikaci dispečerského systému s uživateli pomocí SMS zpráv. V případě, že chceme se systémem Reliance spojit zařízení nepracující na některém z podporovaných driverů, lze použít obecný driver tzv. Generic driver. Avšak lze ho použít pouze pod podmínkou znalosti komunikačního protokolu daného zařízení. Seznam nativních komunikačních driverů je uveden zde[7]:

- TECO
- AMiT
- Modbus
- SAUTER EY2400
- Rittmeyer WSR3000
- PROMOS QQ
- INMAT 51, 66
- ELCOR-94
- ELGAS 2
- WAGO
- M-Bus
- BACnet
- Siemens S7
- IEC 60870-5-104
- IEC 62056-21
- CIMON
- Modicon
- Johnson Controls 9100, FX
- SMS
- Generic

### OPC a DDE servery

Další možností komunikace jsou OPC a DDE servery. Jedná se o celosvětové standardy pro spojení HW se SW různých výrobců. Systém Reliance má modul OPC klienta, který je schopný se připojit k jakémukoliv OPC serveru. OPC servery lze využít ve stejném případě, jako když využíváme Generic Driver, a to tedy pokud nemáme dostupný nativní komunikační driver[8].

### 2.1.3 Licenční a cenová politika

Systém Reliance je distribuován jako Shareware licence, jež nese omezení 25 datových bodů pro zkušební použití zdarma. Pokud mluvíme o datovém bodu, pak se jedná zpravidla o analogovou nebo binární proměnnou. U jednoduchých proměnných typu například BOOL jedna proměnná je rovna jednomu datovanému bodu. U proměnných s větším množstvím prvků se nám započítává jako datový bod každý pátý prvek, například proměnná typu pole.

V případě, že využíváme jen virtuální proměnné, pak se nám tyto proměnné nezapočítávají do celkového počtu 25 datových bodů. Tato zkušební licence neslouží k trvalému běhu, ale pouze k testovacím účelům.

K plné verzi programu je potřeba zakoupit si licenci. Licenční verze s maximálním počtem 200 datových bodů jsou dodávány se SW klíčem. V případě licencí větších než 200 datových bodů je licence chráněna HW klíčem v podobě USB nebo LPT. Na přání zákazníka lze místo SW klíče dodat licenci na HW klíči za poplatek 1 300 Kč. Tento krok umožňuje provozovat licenci na více PC stanicích oproti SW klíči, který je určen jen pro jednu stanicí. Veškeré ceny jsou uvedeny bez DPH[9].

#### 2.1.3.1 Vývojové prostředí

Zde máme na výběr z verzí Design Desktop a Design Enterprise, jak už bylo uvedeno výše verze Enterprise podporuje síťové aplikace s více PC stanicemi na rozdíl od verze Desktop. Od toho se také odvíjí jejich cena, která je uvedena v *Tab. 2-1*.



Tab. 2-1 Srovnání cen vývojového prostředí Reliance 4 Design verzí Desktop a Enterprise[9]

Počet bodů	Desktop	Enterprise
25	zdarma	-
100	1 900 Kč	2 900 Kč
200	2 900 Kč	3 900 Kč
250	4 900 Kč	6 900 Kč
500	5 900 Kč	8 900 Kč
1 000	9 900 Kč	15 900 Kč
3 000	15 900 Kč	21 900 Kč
5 000	19 900 Kč	25 900 Kč
10 000	25 900 Kč	35 900 Kč
nad 10 000	35 900 Kč	55 900 Kč

### 2.1.3.2 Běhové prostředí (runtime moduly)

Jednotlivé běhové moduly se prodávají samostatně s obdobnou cenovou politikou jako jednotlivé verze vývojového prostředí. Přehled cen jednotlivých modulů v závislosti na počtu datových bodů je uveden v následující tabulce:

Tab. 2-2 Srovnání cen běhového prostředí Reliance 4 View, Control, Server, Control Server[9]

Počet bodů	View	Control	Server	Control Server
25	zdarma	zdarma	zdarma	zdarma
50	1 100 Kč	1 900 Kč	2 200 Kč	2 500 Kč
100	2 200 Kč	4 400 Kč	4 600 Kč	4 900 Kč
200	3 500 Kč	5 400 Kč	5 600 Kč	5 900 Kč
250	5 900 Kč	14 900 Kč	14 900 Kč	17 500 Kč
500	10 900 Kč	20 900 Kč	20 900 Kč	24 000 Kč
1 000	14 900 Kč	29 000 Kč	29 000 Kč	33 000 Kč
3 000	22 000 Kč	39 000 Kč	39 000 Kč	46 000 Kč
5 000	31 000 Kč	59 000 Kč	59 000 Kč	69 000 Kč
8 000	39 000 Kč	69 000 Kč	69 000 Kč	78 000 Kč
10 000	45 000 Kč	79 000 Kč	79 000 Kč	95 000 Kč
nad 10 000	52 000 Kč	99 000 Kč	99 000 Kč	115 000 Kč

### 2.1.3.3 Kombinace vývojového a běhového prostředí

Tento typ licence je vhodný pro vývojáře, kteří mají své zákazníky ve velké vzdálenosti a potřebují mu nějakou část instalace opravit přes vzdálenou plochu, kdy se připojí a upraví projekt přímo u zákazníka. Tato licence je ve dvou verzích a to Combi Package Desktop, která je kombinací modulů Design Desktop, Control a View.

Druhá verze je Combi Package Enterprise, složená z kombinace modulů Design Enterprise, View, Control, Server a Control Server. V ceně této verze je zahrnuta licence pro jednoho tenkého klienta.

Cena těchto balíčků je velmi příznivá, v podstatě jsou tyto balíčky jen o 10 % dražší než nejdražší runtime modul daného balíčku. A za tuto cenu získáme další dva moduly pro verzi Desktop a čtyři pro Enterprise navíc. Ceny těchto balíčků jsou uvedeny v následující tabulce[9]:

Tab. 2-3 Srovnání cen balíčků Reliance 4 Combi Package Desktop a Enterprise[9]

Počet bodů	Desktop	Enterprise
250	16 500 Kč	19 300 Kč
500	23 000 Kč	26 500 Kč
1 000	32 000 Kč	36 300 Kč
3 000	43 000 Kč	50 600 Kč
5 000	65 000 Kč	75 900 Kč
8 000	76 000 Kč	85 800 Kč
10 000	87 000 Kč	104 500 Kč
nad 10 000	109 000 Kč	126 500 Kč

#### 2.1.3.4 Tenčí klienti

Systém Reliance má celkem tři tenké klienty a to Web Client, Smart Client a Mobile Client. V licencích pro tenké klienty není rozdíl, jelikož všechny pracují na webovém rozhraní (není potřeba jejich instalace)[9].

Tab. 2-4 Množstevní ceník tenkých klientů[9]

Počet licencí	Cena
1	V ceně datového serveru
2	9 000 Kč
3	13 500 Kč
5	19 500 Kč
10	28 300 Kč
25	44 000 Kč
50	56 800 Kč

#### 2.1.3.5 Komunikační drivery a OPC klient

Systém Reliance podporuje celou řadu komunikačních driverů. Jako nejvhodnější je použití výrobků společnosti TECO a.s., jelikož se jedná o dceřinou společnost společnosti GEOVAP, spol. s.r.o. a jejich komunikační driver je zdarma. Ostatní komunikační drivery jsou zpoplatněny dle počtu datových bodů. V případě 200 datových bodů je cena u většiny driverů

1 250 Kč, případně nižší. Pokud je požadavek na větší počet datových bodů, pak je cena u většiny 12 500 Kč, či nižší a jejich počet není již omezen.

Modul OPC klienta je součástí instalačního balíčku a není zpoplatněn. OPC servery jsou zpoplatněny a společnost nabízí na svých internetových stránkách přístup k nabízeným serverům za ceny 13 300 Kč a 15 000 Kč[9].

### **2.1.3.6 Technická podpora a další nabízené služby.**

Technická podpora je zdarma pro účastníky školení programu Reliance, případně také pro ty, kteří si pořídili výukový videozáznam tohoto školení. V opačném případě je interní technická podpora zpoplatněna 750 Kč/h, případně externí 900 Kč/h.

Zmíněné školení je rozděleno do 4 výukových dnů, cena jednoho výukového dne je 3 500 Kč, cena jednoho dne záznamu školení je 3 000 Kč[9].

## **2.2 Promotic 8.3**

Systém Promotic je opět SW české produkce a to konkrétně společnosti MICROSYS, spol. s.r.o. Stejně jako systém Reliance sahá historie vývoje Promoticu do roku 1991. V tomto roce byl uvolněn systém pod názvem Promotic DOS. Od roku 1996 je Promotic pod platformou Windows. V současné době je dostupná stabilní verze Promotic 8.3[10].

### **2.2.1 Využití**

Jedná se opět o značně rozšířený systém jak u nás tak po celém světě s více jak tisíci jeho instalací. Z významnější instalací můžeme zmínit například komplexní ovládací a řídicí systém letiště Marnitogorsk, terminál elektrárny Mělník 1, řídicí systémy pro Povodí Vltavy s.p. a monitorovací a řídicí systém experimentálního zplyňovacího zařízení Technické univerzity Košice[3], [11].

### **2.2.2 Struktura systému**

Systém Promotic je tvořen jednotlivými softwarovými moduly. Zjednodušeně je lze rozdělit na vývojové prostředí, běhové prostředí (runtime moduly), tenké klienty, nativní komunikační drivery, OPC a DDE klienta[12].

#### **2.2.2.1 Vývojové prostředí - PmDevelop**

Stejně jako modul Design v systému Reliance slouží tento modul k vytvoření vizualizačního prostředí a struktury celého projektu. Je rozdělen do dvou základních částí.

První částí je editor aplikace, který je základním nástrojem tvorby projektů. Slouží k nadefinování stromové struktury objektů, nastavení těchto objektů, programování skriptů, atd. K programování skriptů se využívá stejně jako u systému Reliance skriptovacího jazyku VBScript.

Druhou částí modulu je editor obrazů, který slouží k vytvoření grafického prostředí aplikace. Grafické prostředí se vytváří skládáním skupin jednotlivých grafických prvků, které jsou předem nadefinovány. U těchto prvků lze definovat statické hodnoty a v případě propojení na datovou vazbu vytvořit vizualizace.

Součástí tohoto modulu je také informační a diagnostický INFO systém, který dovoluje prohlížení důležitých informací za běhu aplikace[12].

### **2.2.2.2 Běhové prostředí (runtime moduly) - PmRuntime**

Jedná se o běhové prostředí, jeho užitnou hodnotu můžeme přirovnat k modulu Reliance Control. Tento modul umožňuje dispečerské řízení, grafické zobrazování dat, zobrazování alarmů a jejich kvitování, zaznamenávání akcí operátora. Dále umožňuje záznam systémových dat s časovým údajem, ze kterých lze vytvářet tabulky či grafy. Také je zde možnost přihlášení a odhlášení jednotlivých uživatelů[13].

### **2.2.2.3 Tencí klienti**

Mezi zástupce tenkých klientů patří pouze dva následující moduly.

#### **PmDataClient**

Tento klient slouží k získávání a zasílání dat pomocí standardu XML. Zejména komunikace mezi objekty, čtení XML dat z objektů – alarmů, uživatelských zásahů, dat pro tvorbu tabulek, či grafů[14].

#### **PmWebClient**

Web klient umožňuje zobrazení grafického prostředí pomocí internetového prohlížeče. Od verze 8.2 lze využít tento klient v jakémkoliv prohlížeči a také v tabletech a chytrých telefonech, které nabývají čím dál větší obliby. Grafické rozhraní lze spustit v prohlížeči Internet explorer bez problémů. V případě, že chceme grafické rozhraní spustit v alternativním prohlížeči, je nutné převést skripty z jazyka VBScript do JavaScriptu. Tyto jazyky jsou syntakticky blízce příbuzné, proto by neměl tento fakt být závažným problémem pro vývojáře[14].

### **2.2.2.4 Možnosti komunikace**

Komunikační možnosti jsou zde stejné jako u systému Reliance, lze tedy využít nativních komunikačních driverů, OPC a DDE serverů, nebo webové služby.

#### **Nativní komunikační drivery**

Systém Promotic nepodporuje všechny komunikační drivery, které podporuje systém Reliance. Ale oproti tomu je zde zase podpora jiných komunikačních driverů, které systém Reliance nepodporuje. Pokud srovnáme počty podporovaných driverů těchto dvou systémů, tak je

počet přibližně stejný a je otázka, který konkrétní driver potřebujeme v dané aplikaci. Jsou zde zastoupeny jak obecné komunikační protokoly, tak nejznámějších výrobců PLC. Stejně jako u minulého systému, je zde také protokol pro přijímání a odesílání SMS zpráv.

Pokud systém nemá dostupný komunikační driver pro zařízení, které potřebujeme k systému připojit, můžeme využít univerzálního komunikačního driveru Char, případně si můžeme vytvořit ActiveX objekt, kterým tuto komunikaci vyřešíme. ActiveX objekt se využívá například ke komunikaci s automaty firem AMiT a Johnson Controls. Další možností komunikace je OPC server, nebo kontaktování firmy MICROSYS, které na svých stránkách nabízí možnost dohody na tvorbě dalšího komunikačního driveru.

Seznam nativních komunikačních driverů je uveden zde[15]:

- TECO
- Modbus
- PROMOS RT
- INMAT 51, 66
- ELGAS 2
- M-Bus
- BACnet
- Siemens S5, S7, 3964
- IEC 60870-5-101
- IEC 60870-5-104
- IEC 62056-21
- SNMP
- Bradley DF1, CIP
- Melsec
- S-Bus
- Adam
- Koyo
- Fatek
- Omron
- Teleperm
- MicroUnit
- NET0
- SMS
- Char

## OPC a DDE servery

Jak už bylo uvedeno výše, jednou z možností komunikace jsou OPC a DDE servery, což jsou standardizovaná rozhraní pro komunikaci mezi SW a HW různých výrobců. Součástí tohoto systému je OPC a DDE klient, který umožňuje se připojovat k OPC a DDE serverům[16].

### 2.2.3 Licenční a cenová politika

Obdobně jako systém Reliance, má systém Promotic Shareware licenci, která dovoluje systém využívat s určitým omezením zdarma. Zde je omezení ve formě maximálního počtu proměnných a to počtem 100 pro vývojové prostředí a 30 pro běhové prostředí.

V případě použití většího počtu proměnných je nutné zakoupit licenci. Licence je pro vývojové prostředí dodávána výhradně jako HW klíč na USB případně LPT. U běhových prostředí je nejběžněji dodávána licence v podobě HW klíče, ale lze také pořídit SW klíč, který je vázaný na HW komponenty PC stanice, případně na licenční server.

Veškeré ceny jsou uvedeny bez DPH[17].

#### 2.2.3.1 Vývojové prostředí

Vývojové prostředí má pouze dvě licence. První licence je zdarma a má omezení 100 proměnných. Placená licence není omezena počtem proměnných a její cena je 9 000 Kč. Cena této licence je velmi výhodná oproti systému Reliance, kde za tuto cenu pořídíme licenci na 1000 datových bodů.

Speciální licence je uvedena pro školy a její cena je 1000 Kč. Dále je možné zakoupení licence na ochranu aplikace jedinečným číslem uživatele, jejíž cena je 500 Kč[17].

### 2.2.3.2 Běhové prostředí (runtime moduly)

Běhové prostředí má čtyři základní licence. Tyto licence jsou rozdílné počtem proměnných a jsou přehledně zobrazeny v (Tab. 2-5). Sloupec označený jako Sleva uvádí slevy na licence komunikačních driverů, standardních rozhraní a přístupu k serveru.

Ceny těchto licencí jsou opět velmi výhodné oproti systému Reliance, kdy za cenu nejdražší licence pořídíme licenci systému Reliance v rozmezí 1000 – 3000 datových bodů[17].

Tab. 2-5 Srovnání cen běhového prostředí dle počtu proměnných[17]

Počet proměnných	Sleva	Cena
30	-	zdarma
100	70%	5 000 Kč
50000	-	20 000 Kč
nad 50000	-	40 000 Kč

### 2.2.3.3 Tencí klienti

License pro tenké klienty se prodává zvlášť pro data a web klient. Licence Web klienta zahrnuje vlastnosti Data klienta. Ceny licencí se odvíjejí od jejich počtu a jsou uvedeny v (Tab. 2-6)[17].

Ceny Web klienta jsou srovnatelné s cenami tenkých klientů systému Reliance.

Tab. 2-6 Množstevní ceník tenkých klientů[17]

Klient	Počet licencí	Cena
Data	Každý 1-3	1 000 Kč
	každá další	500 Kč
Web	Každý 1-2	5 000 Kč
	Každý 3-5	2 500 Kč
	Každý 6-50	1 000 Kč

### 2.2.3.4 Komunikační drivery a OPC klient

Cena komunikačních driverů se pohybuje v rozmezí 5 000 Kč až 8 000 Kč. Jediný komunikační driver stojí 20 000 Kč.

Oproti systému Reliance není zde žádné omezení na počet proměnných (datových bodů), proto pro větší aplikace bude cenově výhodnější systém Promotic a pro menší Reliance.

Modul OPC klienta je zde zpoplatněn částkou 4 000 Kč[17].

### 2.2.3.5 Technická podpora a další nabízené služby

Krátké konzultace prostřednictvím emailu či telefonu u tohoto systému jsou zdarma, zpoplatněny jsou delší interní konzultace ohledně systému Promotic částkou 700 Kč/h, a interní konzultace nad aplikací zákazníka částkou 1 000 Kč/h. Konzultace přes vzdálenou obrazovku je zpoplatněna částkou 1 500 Kč.

Podobně jako u systému Reliance je zde nabízeno školení, s cenou 4 500 Kč.

Dále je u systému Promotic zpoplatněn přístup k externím databázím, externím DLL knihovnám a zabezpečenému HTTPS rozhraní pro Web klienta[17].

## 2.3 TIRS.NET 6

Systém TIRS.NET je dalším českým zástupcem ve světě SCADA/HMI systémů. Je to produkt firmy CORAL s.r.o., která pracuje na vývoji SCADA/HMI systémů již od roku 1992. Předchůdci toho systému jsou TIRS32 a TIRSWeb[18].

### 2.3.1 Využití

Tento systém je vhodný jak pro rozsáhlejší aplikace, tak i pro menší projekty. Jako rozsáhlejší aplikace můžeme uvést například řídicí systémy pro Povodí Labe s.p. s pěti hlavními servery a 43 podřízenými dispečinkami nebo České Radiokomunikace a.s. s šesti oblastními servery pro monitorování a řízení vysílačů.

Dalšími aplikacemi jsou například monitoring a řízení rozvodů a předávacích stanic plynu Jihomoravské plynárenské, a.s. a monitoring a řízení vytápění, vzduchotechniky a osvětlení Veletrhy Brno, a.s[3], [19].

### 2.3.2 Struktura systému

Tento systém má odlišnou koncepci nežli dva předchozí systémy. Logicky jeho koncepci lze rozdělit na dvě oddělené části.

První část serverová, které je pro uživatele neviditelná, jelikož běží jako služba Windows. Do této části spadá jádro systému, komunikační konektory a databázové moduly. Tato serverová část se spouští automaticky ihned po zapnutí PC stanice a startu systému Windows, bez nutnosti přihlášení uživatele.

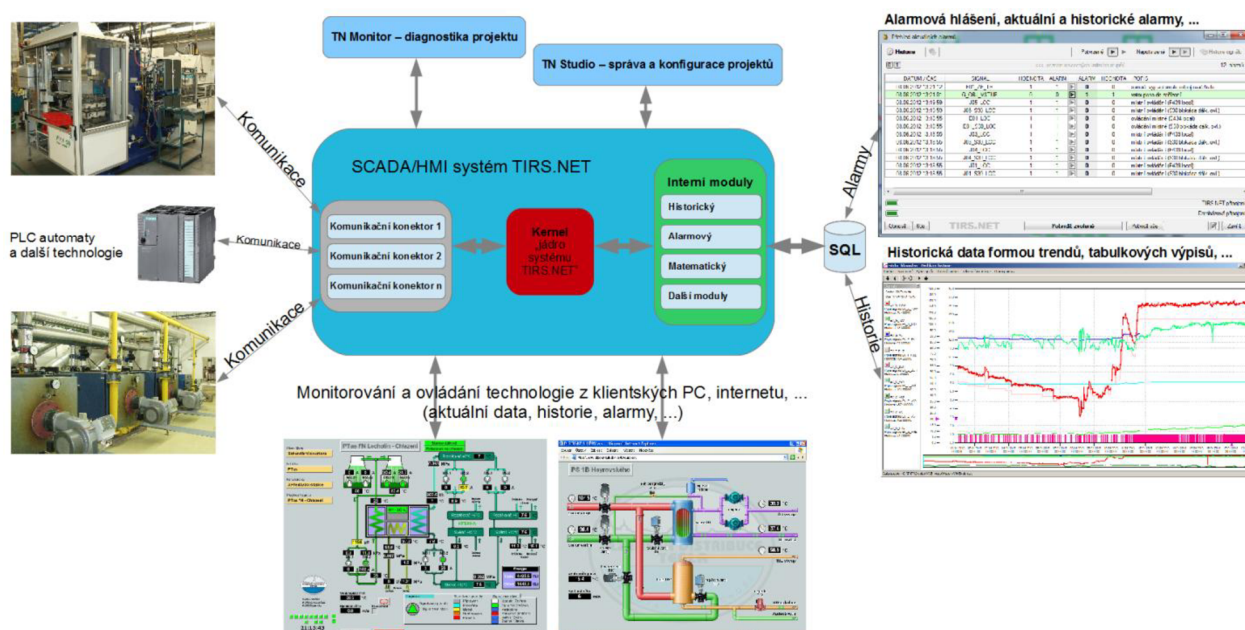
Druhá část klientská, je viditelná pro uživatele. Spadá sem obrazovka s vizualizací připojené technologie, alarmy, historická data, diagnostické a konfigurační moduly.

Koncepcí systému je zobrazena na (Obr. 2-1), kde je názorně rozdělena klientská a serverová část.

Oba předchozí systémy byly rozděleny na vývojové a běhové prostředí. Obě tyto prostředí jsou zde sloučené do jedné aplikace[20], [21].

### 2.3.2.1 Vývojové a běhové prostředí

Jak už bylo uvedeno, tento systém má sloučenou funkci vývojového a běhového prostředí do jedné aplikace. Viditelnou částí aplikace je například modul TN Panel, který dovoluje oprávněnému uživateli vytvořit či editovat grafické rozhraní připojené technologie. K prohlížení těchto vytvořených grafických rozhraní slouží modul TN Viewer. Vytvoření a editace grafického rozhraní je možné z libovolné, klientské PC stanice, a je pouze závislé na oprávnění daného uživatele. [20], [21].



Obr. 2-1 Koncepce systému TIRS.NET[20]

### 2.3.2.2 Tenci klienti – TN WEB

Jedná se o webovou nadstavbu, která umožňuje přenesení celé aplikace do téměř libovolného internetového prohlížeče. Podmínkou pro funkčnost této nadstavby je nutná komponenta Microsoft Silverlight, která tvoří dynamický online obsah a umožňuje interaktivní práci s ním. Tato komponenta je určena pro OS Windows a Mac OS X. Pro platformy Linux je k dispozici obdobná komponenta pod názvem Moonlight.

Tento tenký klient nám opět jako u předešlých systémů dovoluje monitorovat a řídit danou technologii. Monitorovat grafy, videa, alarmy, atd.[22].

### 2.3.2.3 Možnosti komunikace

Komunikační možnosti jsou zde obdobné jako u obou předchozích systémů. Ke komunikaci jsou k dispozici vlastní moduly, tzv. komunikační konektory. Tyto konektory využívají nativní protokoly. Dále lze využít OPC a DDE serverů a webové služby[23].



## Komunikační konektory

Škála komunikačních konektorů je zde početně o něco málo větší než u předchozích systémů. Jsou tady opět zastoupeny protokoly nejznámějších výrobců PLC a obecné protokoly. I u tohoto systému je komunikační protokol k přijímání a odesílání SMS zpráv. Seznam nejčastěji používaných komunikačních konektorů je uveden zde[23]:

- TECO
- Modbus
- INMAT 51, 66
- ELCOR-94
- ELGAS 2
- M-Bus
- BACnet
- IEC 60870-5-101
- IEC 60870-5-104
- SNMP
- Bradley DF1
- S-Bus
- Adam
- AMiT
- PROMOS QQ
- Siemens S7
- CometMS
- FM\_MS
- Hitachi
- LGtool
- Megabus
- MICROPEL\_EPNP
- MIKROT
- JohnsonControls N2BUS
- NOEL3000
- RMMS
- SAUTER EY2400, 3600
- TOPOL
- TREND
- Ametek
- AvalonGTW
- Heraeus
- HW group
- T-M505
- TEC2000
- SMS

## OPC a DDE servery

V případě, že není dostupný komunikační konektor je možnost využít OPC a DDE servery obdobně jako u předchozích systémů. Je tedy možné k systému připojit jakékoliv zařízení, ke kterému je k dispozici OPC server[23].

### 2.3.3 Licenční a cenová politika

I tento systém je k dispozici zkušební verzi. Jedná se o DEMO verzi, která je časově omezena. Současně s touto verzí jsou k dispozici zkušební projekty.

Veškeré ceny jsou uvedeny bez DPH[24].

#### 2.3.3.1 Vývojové a běhové prostředí – TIRS.NET Base

Tento systém, jak už bylo uvedeno, není rozdělen na vývojovou a běhovou verzi a je distribuován společně jako celek. Politika výrobce je taková, že vývojová licence je pro implementační a realizační firmy, výrobce a dodavatele zařízení nebo komponent zdarma. Důvod, proč výrobce dává svůj systém zdarma vývojářům, je patrně ten, aby posílil své postavení na trhu. Je to určitá forma reklamy, kterou s největší pravděpodobností zaplatí koncový uživatel.

Pro koncové uživatele se cena systému skládá ze základní sady označené TIRS.NET Base, která je určena k instalaci na jednu PC stanici, a z ceny dalších klientů, o které můžeme systém rozšířit. Cena jak základní sady, tak klientů je závislá na počtu datových bodů. Ceny tohoto systému i klientů jsou uvedeny v následující tabulce[24]:

Tab. 2-7 Srovnání cen systému TIRS.NET Base v závislosti na počtu datových bodů[24]

Počet bodů	základní sada	každý 1-10 klient	každý další klient
25	3 800 Kč	1 400 Kč	1 100 Kč
50	4 600 Kč	1 600 Kč	1 400 Kč
100	6 200 Kč	2 200 Kč	1 600 Kč
200	9 600 Kč	2 700 Kč	2 200 Kč
500	13 100 Kč	3 400 Kč	2 700 Kč
1000	16 600 Kč	4 000 Kč	3 400 Kč
3000	20 800 Kč	5 400 Kč	4 700 Kč
6000	41 600 Kč	6 700 Kč	5 400 Kč

### 2.3.3.2 Tencí klienti

Webová nadstavba TN WEB systému je zpoplatněna 100 % ceny TIRS.NET Base[24].

### 2.3.3.3 Komunikační konektory a OPC servery

Komunikační konektory jsou rozděleny do dvou skupin. Jejich cena je určena procentuálně z ceny základní sady TIRS.NET Base, a to 35 % ceny pro první skupinu a 55 % pro druhou. Z procentuálního určení ceny vyplývá, že u velmi rozsáhlých systémů, co do počtu datových bodů a klientů, budou komunikační konektory poměrně nákladné oproti oběma předchozím systémům.

OPC klient je součástí první skupiny a jeho cena je tedy tvořena 35 % ceny základní sady[24].

### 2.3.3.4 Technická podpora a další nabízené služby

Podobně jako u systému Promotic je i zde nabízena obecná podpora zdarma. Dále je nabízena podpora s garantovanou dobou odezvy. Pro tuto podporu je nutné mít se společností CORAL s.r.o., uzavřenu smlouvu, cenové podmínky nejsou zde uvedeny.

Školení je pro implementační a realizační firmy, výrobce a dodavatele zařízení nebo komponent zdarma stejně jako základní sada TIRS.NET Base.

Dále jsou nabízeny moduly pro rozesílání alarmových hlášek prostřednictvím SMS či emailem, každý tento modul je zpoplatněn cenou 5 400 Kč. Případně je také dostupný modul, který rozesílá vybrané signály na nadřazené dispečinky přes internet. Cena toho modulu je 16 000 Kč[24].

## 2.4 Výsledné srovnání

Srovnání uvedených systémů, by bylo nejvhodnější uvést na modelovém příkladu. Vytvoření modelové situace by nemuselo být objektivní, jelikož by se model mohl přiklánět nastavenými parametry k jedné či jiné variantě. Pro názornost byly systémy srovnány v následujících

tabulkách. V (Tab. 2-8) jsou uvedeny ceny jednotlivých systémů vývojového prostředí v závislosti na počtu datových bodů. Z této tabulky plyne, že systém TIRS.NET je nejvýhodnější pro vývojáře. V (Tab. 2-9) jsou uvedeny ceny jednotlivých systémů běhových prostředí v závislosti na počtu DB. Z této tabulky je patrné, že systém Reliance 4 je poměrně dražší než zbývající dva systémy. V (Tab. 2-10) jsou uvedeny ceny tenkých klientů v závislosti na jejich počtu. Z uvedených tabulek je zřejmé, že systémy Reliance 4 a Promotic 8.3 mají podobné ceny za tenké klienty, cena u systému TIRS.NET 6 se odvíjí od počtu DB. V (Tab. 2-11) jsou uvedeny ceny nejpočetnější cenové skupiny jednotlivých komunikačních driverů v závislosti na počtu DB. Ceny jsou uvedeny za jeden komunikační driver a jejich cena je u jednotlivých systémů podobná.

Tab. 2-8 Srovnání cen vývojových prostředí jedn. systémů v závislosti na počtu DB[9],[17],[24]

Počet bodů	Reliance 4	Promotic 8.3	TIRS.NET 6
100	2 900 Kč	-	-
200	3 900 Kč	9 000 Kč	-
500	8 900 Kč	9 000 Kč	-
1 000	15 900 Kč	9 000 Kč	-
2 000	21 900 Kč	9 000 Kč	-

Tab. 2-9 Srovnání cen běhových prostředí jedn. systémů v závislosti na počtu DB[9], [17], [24]

Počet bodů	Reliance 4	Promotic 8.3	TIRS.NET 6
100	4 900 Kč	5 000 Kč	6 200 Kč
200	5 900 Kč	20 000 Kč	9 600 Kč
500	24 000 Kč	20 000 Kč	13 100 Kč
1 000	33 000 Kč	20 000 Kč	16 600 Kč
2 000	46 000 Kč	20 000 Kč	20 800 Kč

Tab. 2-10 Srovnání cen tenkých klientů jedn. systémů v závislosti na počtu klientů[9],[17],[24]

Počet klientů	Reliance 4	Promotic 8.3	TIRS.NET 6*
2	9 000 Kč	10 000 Kč	2 800 - 13 400 Kč
5	19 500 Kč	17 500 Kč	7 000 - 33 500 Kč
10	28 300 Kč	22 500 Kč	14 000 - 67 000 Kč

\*ceny jsou uvedeny v rozmezí v závislosti na počtu DB.

Tab. 2-11 Srovnání cen komunikačních driverů jedn. systémů v závisl. na počtu DB[9],[17],[24]

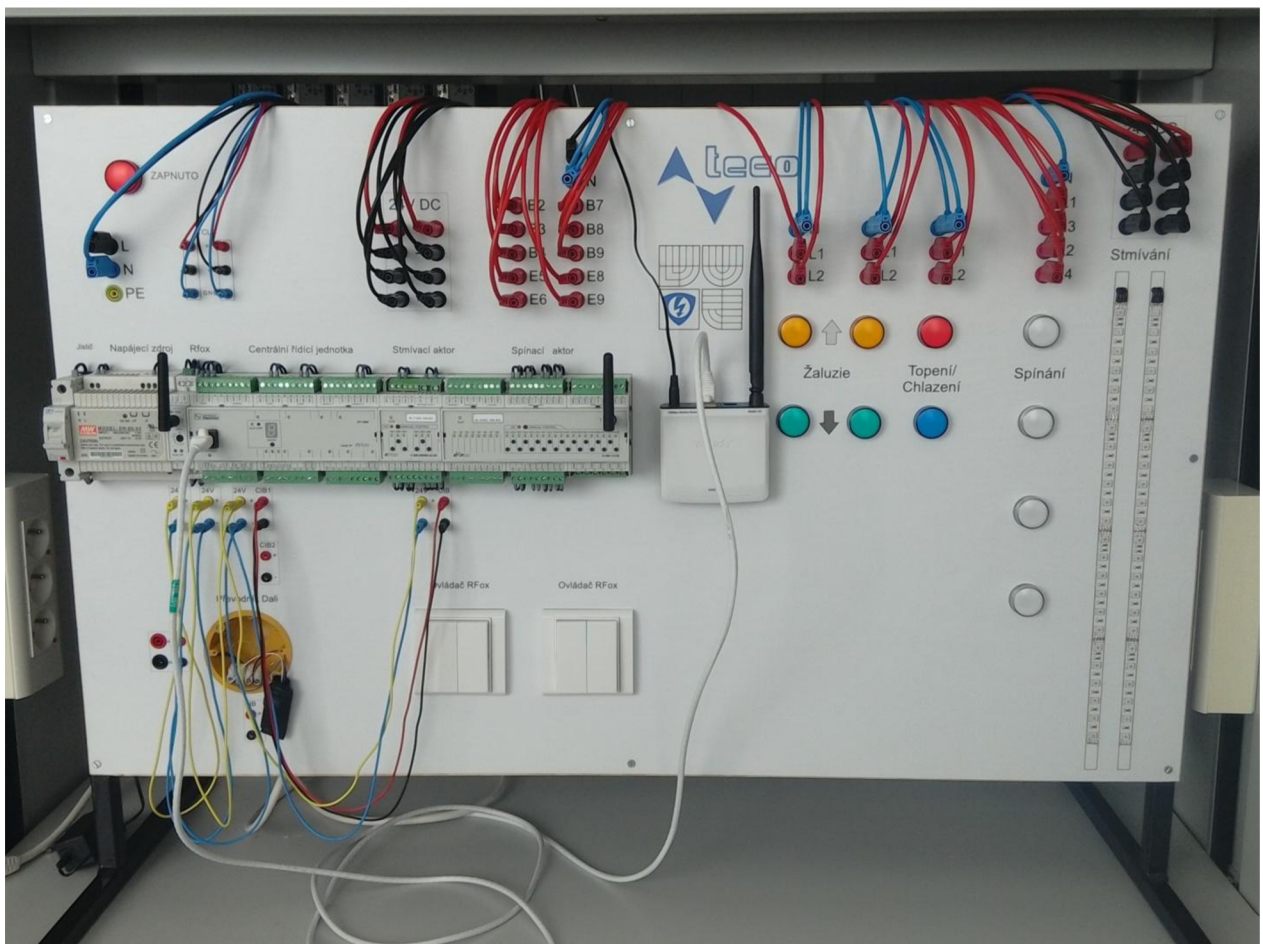
Počet bodů	Reliance 4	Promotic 8.3	TIRS.NET 6**
100	1 250 Kč	2 400 Kč	3 410 Kč
200	1 250 Kč	8 000 Kč	5 280 Kč
500	12 500 Kč	8 000 Kč	7 205 Kč
1000	12 500 Kč	8 000 Kč	9 130 Kč
2000	12 500 Kč	8 000 Kč	11 440 Kč

\*\*skupina č. 2, 55% z ceny běhového prostředí.

### 3 PRAKTICKÁ APLIKACE ŘÍZENÍ REÁLNÉ SYSTÉMOVÉ ELEKTROINSTALACE FOXTROT S VIZUALIZACÍ.

Praktická aplikace vychází z laboratorního panelu Foxtrot na UEEN, který vznikl v rámci diplomové práce Využití řídicího systému Foxtrot jako Building Management System[25]. Panel je zobrazen na (Obr.3.1).

Panel je osazen prvky řady Tecomat Foxtrot společnosti Teco, a.s.. Jedná se o prvky: centrální řídicí jednotka CP 1000, modul řízení LED pásků (C-DM-0006M ULED), komunikační modul RFox (RF-1131), bezdrátový model kombinovaných vstupů a výstupů (R-HM-1121M), bezdrátové skupinové ovladače Time (R-WS-0400R-Time), Převodník na sběrnici DALI (C-DL-0012S), síťový napájecí zdroj (DR-60-24). Dále je zde osazen wi-fi router Tenda.



Obr. 3-1 Laboratorní panel Foxtrot

Nastavení a programování elektroinstalace Foxtrot se provádí v programu Mosaic. Tento program je dostupný ke stažení na internetových stránkách společnosti Teco, a.s.[26]. Program Mosaic je chráněn licenčními podmínkami, jejich ověření probíhá vytvořením komunikace mezi PC, kde je program Mosaic spuštěn, a centrální řídicí jednotkou, kde je HW klíč uložen.

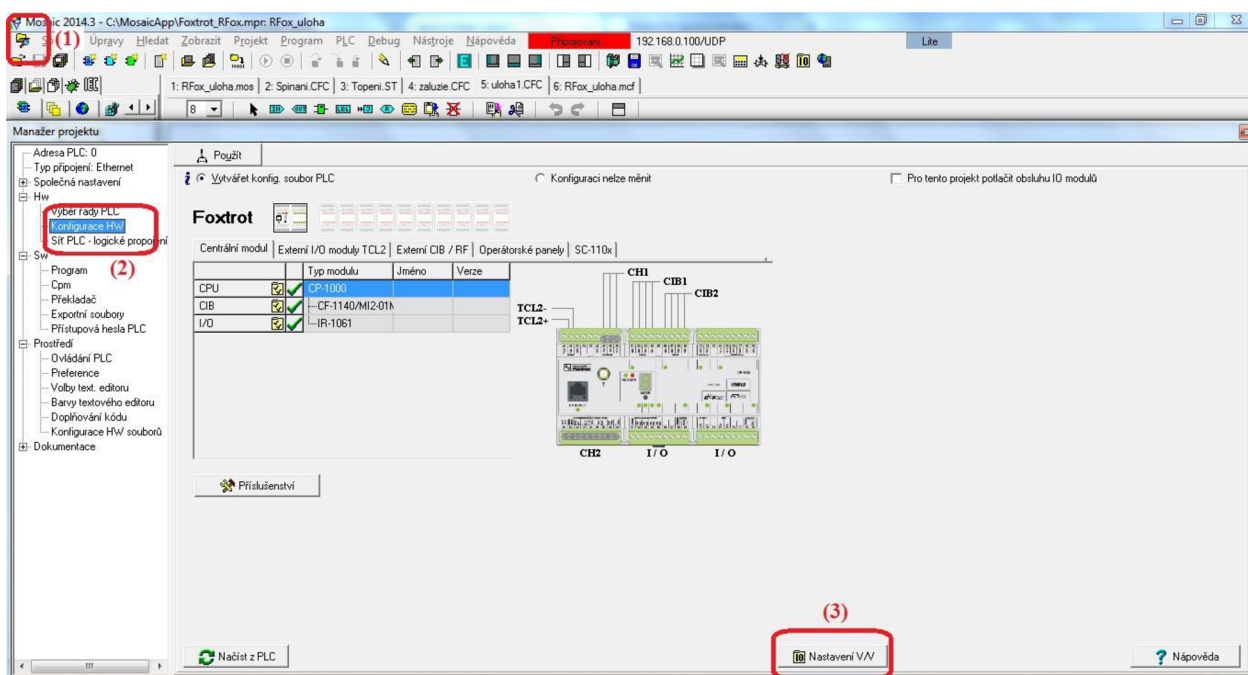
Postup programování elektroinstalace Foxtrot v programu Mosaic je v [25] dostatečně popsán, a proto zde budou popsány jen základní operace v tomto programu. Z [25] byl převzat základní program elektroinstalace Foxtrot, který se zavádí do centrální řídicí jednotky. Převzatý program byl v detailech pozměněn.

K vlastní tvorbě SCADA/HMI vizualizačního prostředí byl využit program Reliance 4.

### 3.1 Nastavení a vygenerování souboru s proměnnými

Pro vzájemnou komunikaci systému Reliance s centrální řídicí jednotkou je nutné z programu Mosaic exportovat proměnné (vstupy a výstupy) v souboru s formátem \*.pub. Z vývojového prostředí Mosaic jsou exportovány proměnné, deklarované jako veřejné {PUBLIC}.

K vyexportování tohoto souboru jsou v programu nejprve provedeny úvodní nastavení a zprovozněna komunikace dle [25]. Po těchto základních nastaveních dle (Obr. 3-2) zobrazíme **Manažera projektu (1)**, následně přepneme na záložku **Konfigurace HW (2)**, a jako třetí krok spustíme modul **Nastavení V/V (3)**

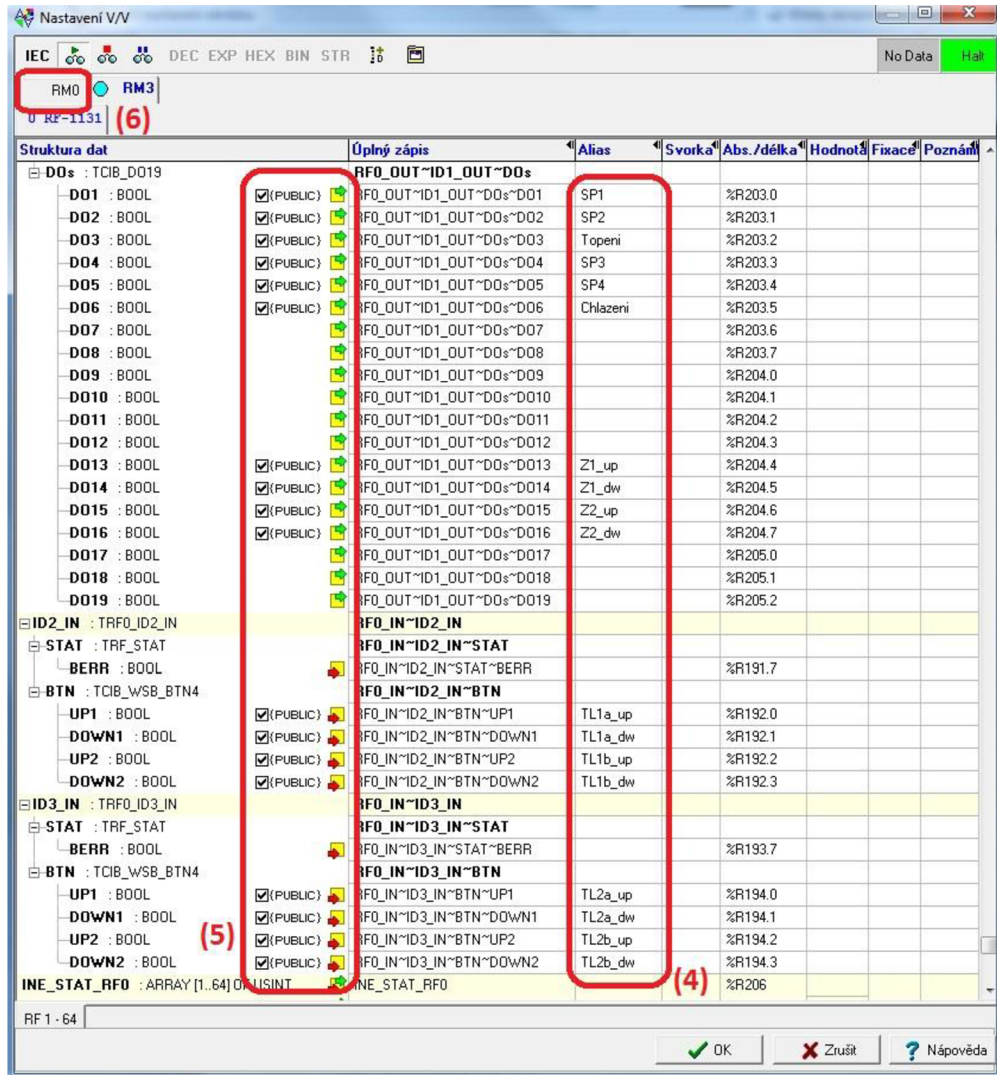


Obr. 3-2 Mosaic - Manažer projektu - Konfigurace HW

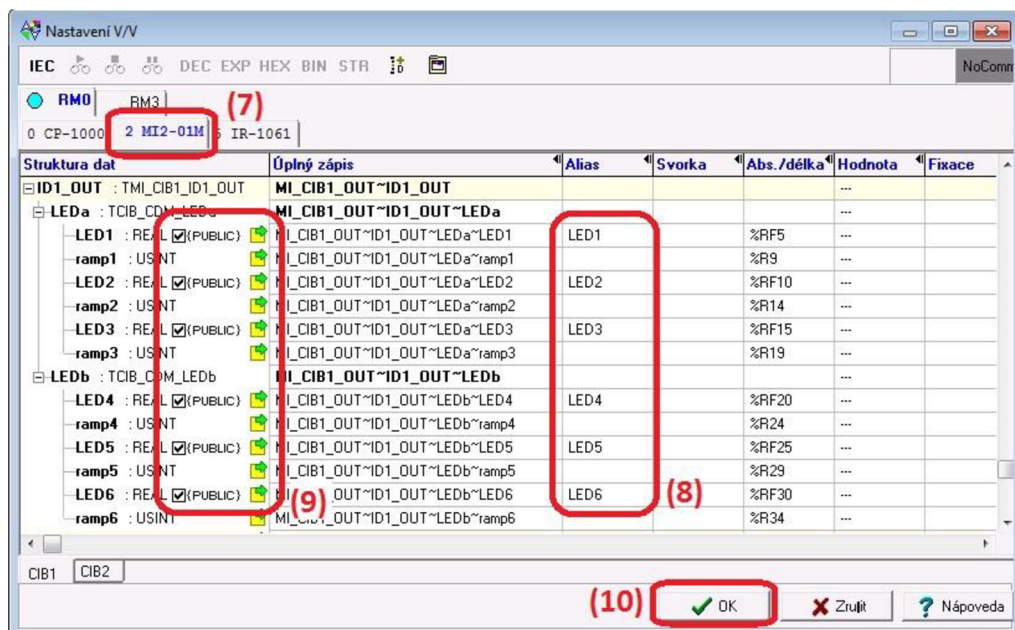
V modulu **Nastavení V/V** (Obr. 3-3) jsou uvedeny proměnné (vstupy a výstupy). Proměnné, které budeme využívat, je nutné **pojmenovat (4)** a **zatrhnout** u nich atribut **{PUBLIC} (5)**.

Po zatrnutí a pojmenování se přepneme na záložku **RM0 (6)**, a dále na (Obr. 3-4) na záložku **MI2-01M (7)**. Zde znovu opakujeme postup s **pojmenováním** proměnných (8) a **zatrnutí** atribut **{PUBLIC} (9)**. Následně modul **Nastavení V/V** potvrdíme volbou **ok (10)**.



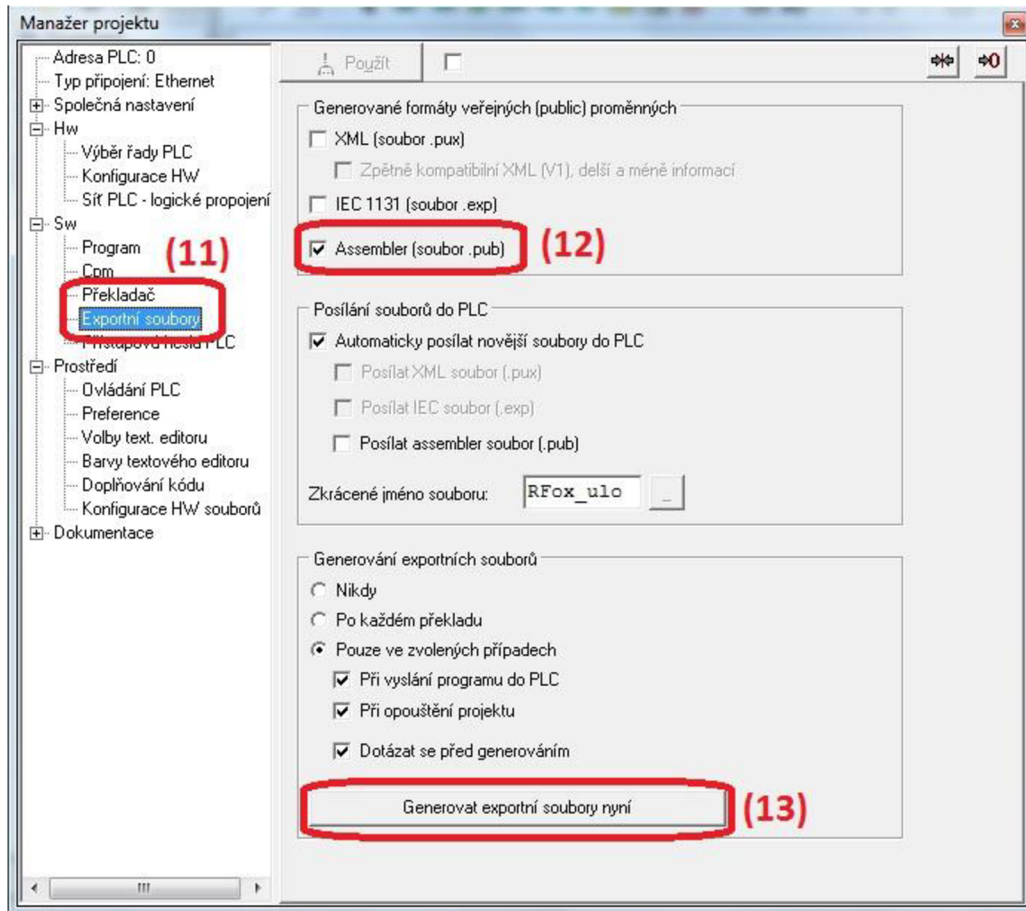


Obr. 3-3 Mosaic - Nastavení V/V – 1



Obr. 3-4 Mosaic - Nastavení V/V – 2

V dalším kroku se v **Manažerovi projektu (1)** (Obr. 3-2) přepneme na záložku **Exportní soubory (11)** (Obr. 3-5), kde zaškrtneme položku **Assembler (soubor .pub) (12)** a dále vygenerujeme exportní soubor pomocí volby **Generovat exportní soubory nyní (13)**. Následně vybereme cestu, kam si přejeme soubor uložit, pojmenujeme ho a uložíme.



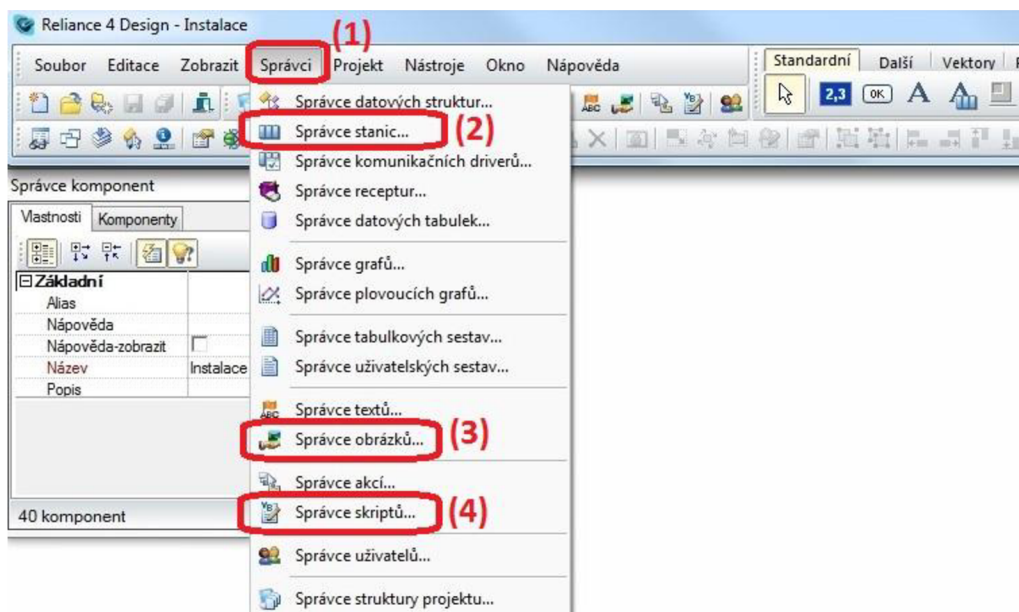
Obr. 3-5 Mosaic - Manažer projektu – Exportní soubory

## 3.2 Vlastní tvorba vizualizačního prostředí

K návrhu vizualizačního prostředí v systému Reliance slouží modul Design, po spuštění tohoto modulu se otevře úvodní obrazovka, na které zvolíme volbu **Vytvořit nový projekt**, následně se otevře průvodce tvorbou nového vizualizačního projektu.

V průvodci vytvoření projektu se nastavují jednotlivé parametry projektu, na další stránku nastavení se pokračuje tlačítkem **Další**. Postupně se nastavuje název projektu, adresář, ve kterém bude projekt uložen, rozlišení vizualizace, kde má uživatel na výběr mezi jednotlivými přednastavenými rozlišeními případně si může zvolit vlastní rozlišení. Dalším nastavovaným parametrem projektu je možnost šifrování projektu zadáním hesla. Toto šifrování je účinné pro vývojové prostředí, v běhovém prostředí není heslo požadováno. Následuje okno s možností uvedení komentáře do projektu. Poslední nastavovaný parametr je výběr, zda chceme vytvořit vizualizační okno, toto potvrdíme a následně se nám otevře průvodce vytvoření nového okna.

V tomto průvodci se opět přesunujeme na další stránky tlačítkem **Další**. Postupně zde zadáváme název vizualizačního okna a titulěk. Na další stránce je možnost zvolit zda okno má být normální či dialogové, a zda má být v okně umístěna lišta, její umístění a velikost.



Obr. 3-6 Reliance Design - Správci

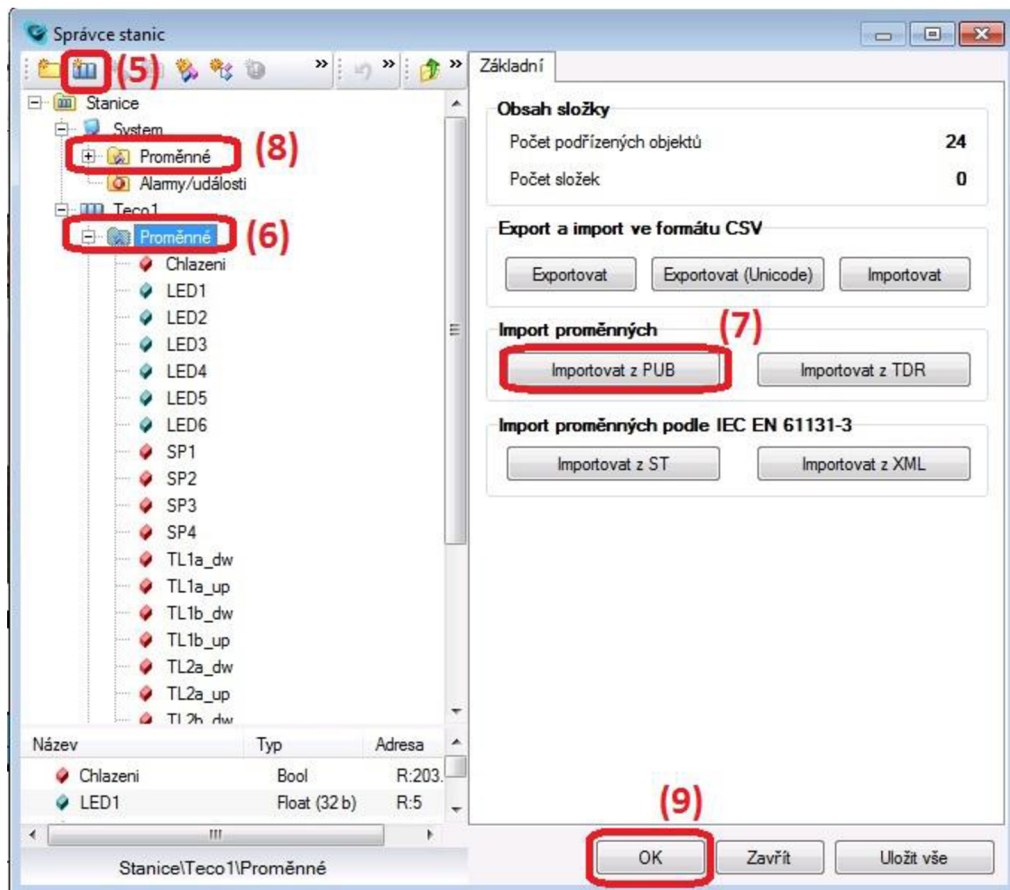
### 3.2.1 Správce stanic

Při tvorbě vizualizačního prostředí je nutné si nejprve definovat proměnné. Definování proměnných se provede v **Správci stanic**, kterého vyvoláme dle (Obr. 3-6) volbou **Hlavní menu / Správci (1) / Správce stanic (2)**.

Ve správci stanic je předdefinována stanice **System**, která slouží k definici vnitřních proměnných, které se nezavádí do žádného zařízení. Dále si zde dle (Obr. 3-7) můžeme přidat pomocí tlačítka **Nová stanice (5)** další stanici dle námi používaného zařízení (driveru). Zvolením této volby se nám zobrazí nabídka stanic, námi vybranou stanicí **Teco1** potvrdíme dvojklikem



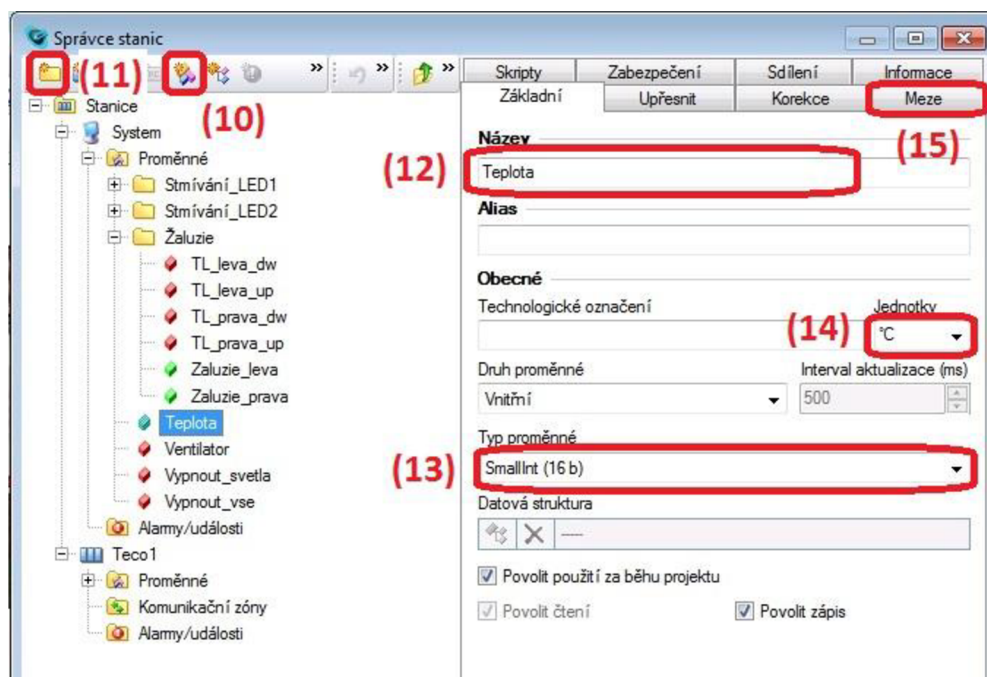
nebo tlačítkem **OK**. U nově vložené stanice **Teco1** se přepneme na záložku **Proměnné (6)**, a v pravé části se nám zobrazí možnost importu proměnných, import provedeme tlačítkem **Importovat z PUB (7)**. Následně vyhledáme cestu k souboru, který jsme si vyexportovali z programu Mosaic a tento soubor otevřeme. Pokud jsme vše provedli správně, pak se nám do stanice **Teco1** načtou proměnné, které mají definované datové typy.



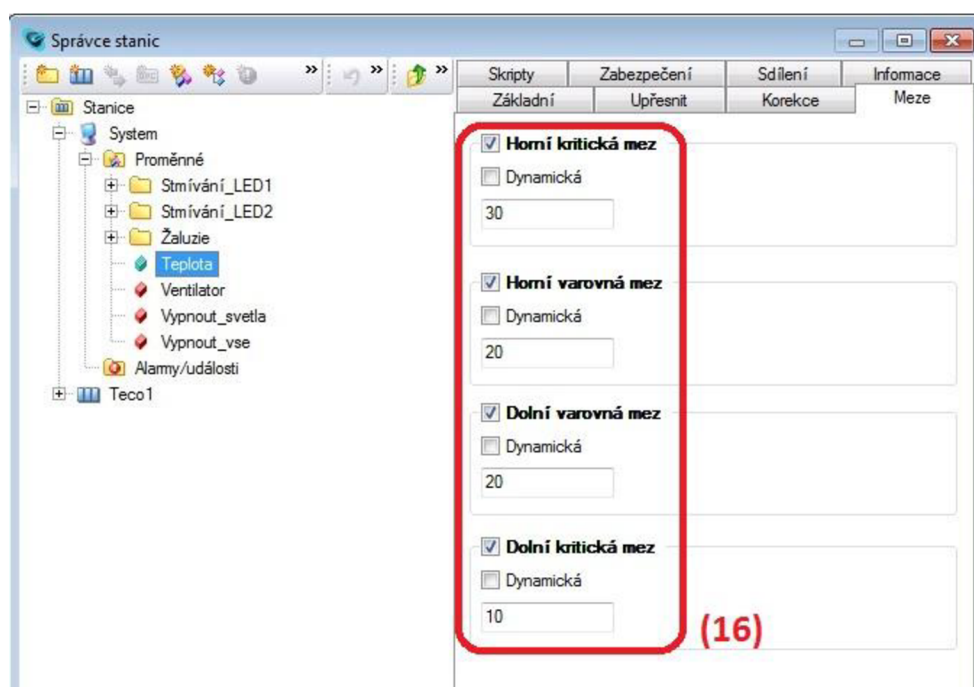
Obr. 3-7 Reliance Design - Správce stanic – 1

Do stanice **System** přidáme vnitřní proměnné, tak že označíme záložku **Proměnné (8)** ve stanici **System** a vložíme je dle (Obr. 3-8) tlačítkem **Nová proměnná (10)**. Případně můžeme proměnné členit do složek, její přidání se provede vytvořením **Nové složky (11)**. Pokud máme označenou nově vytvořenou proměnnou, můžeme upravovat její atributy, např. **Název (12)**, **Typ proměnné (13)**, kde zvolíme požadovaný datový typ. V případě, že zobrazujeme proměnnou ve vizualizaci ve formě číselné hodnoty, pak můžeme u proměnné zobrazit **Jednotku (14)**.

Dále můžeme u proměnných nastavit **Meze (15)**, nastavení mezí je uvedeno na (Obr. 3-9) (16), dle potřeby můžeme u jednotlivých proměnných využít varovné či kritické meze, které mohou být nastaveny na pevnou hodnotu, případně zatrhnutím možnosti **Dynamická**, může být tato mez vázána na jinou proměnnou. Využití mezí je popsáno v (kap. 3.2.4). Ukončení Správce stanic provedeme tlačítkem **Ok (9)**.



Obr. 3-8 Reliance Design - Správce stanic – 2



Obr. 3-9 Reliance Design - Správce stanic – 3

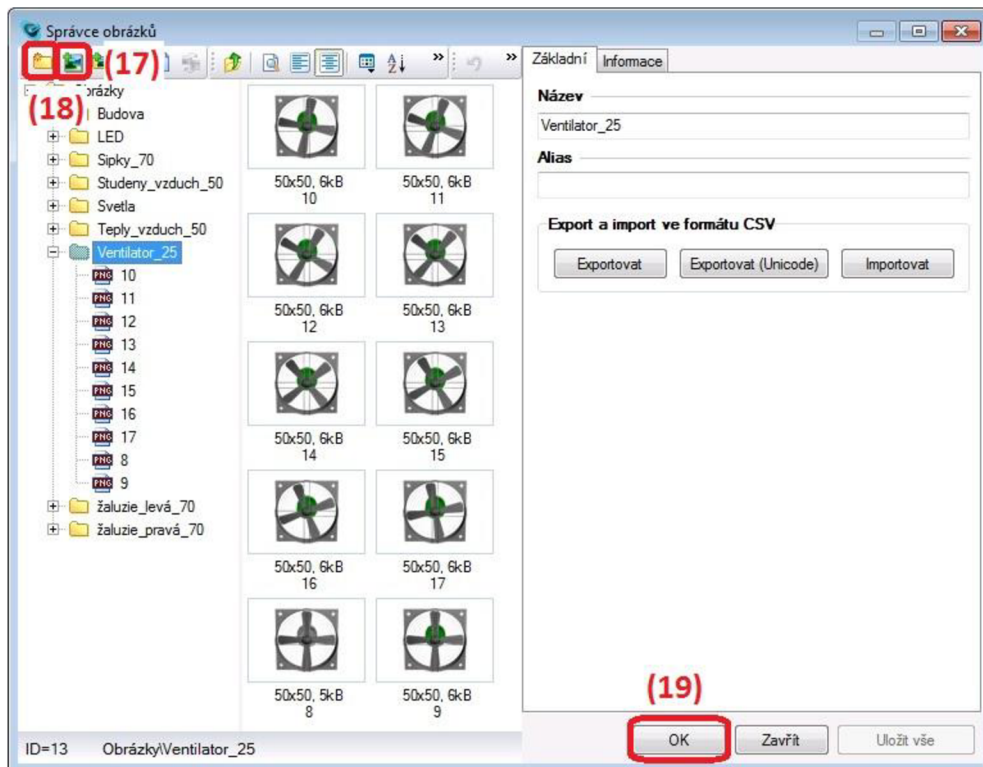
### 3.2.2 Správce obrázků

Při tvorbě grafické části projektu lze využít vlastních obrázků, či obrázků ze systémové knihovny. Správce obrázků spustíme dle (Obr. 3-6) volbou **Hlavní menu / Správci (1) / Správce obrázků (3)**.

Ve správci obrázků lze obrázky přidávat dle (Obr. 3-10) tlačítkem **Přidat obrázky (17)**. Toto tlačítko nás přeměruje do adresáře, kde jsou uloženy obrázky systémové knihovny.

V případě, že knihovnu nemáme staženou, je možnost ji stáhnout v zákaznické sekci na stránkách společnosti po přihlášení, případně stáhnout Trial verzi, kde jsou prvky opatřeny ochranným vodoznakem. Pokud chceme využít možnosti přidání vlastních obrázků či grafických prvků, tak si můžeme po (17), vyhledat cestu do vlastního adresáře a obrázky přidat.

Obdobně jako u správce stanic je i ve správci obrázků možnost členit obrázky do složek pokynem **Nová složka (18)**. Členění do složek je vhodné zejména při rozsáhlejších projektech, nebo v případě, kdy budeme vytvářet animace. Ukončení Správce stanic provedeme tlačítkem **Ok (19)**.



Obr. 3-10 Reliance Design- Správce obrázků

### 3.2.3 Správce skriptů

Jelikož jsou SCADA systémy určeny mimo jiné k řízení procesů, je ve vývojovém prostředí velmi podstatnou funkcí vytváření řídicí logiky. V systému Reliance slouží k této činnosti správce skriptů, který nám umožňuje vytvářet skripty. Správce skriptů spustíme dle (Obr. 3-6) volbou **Hlavní menu / Správci (1) / Správce skriptů (4)**.

Nové skripty lze přidávat dle (Obr. 3-11) tlačítkem **Nový skript (20)**. Obdobně jako u předchozích správců je i ve správci skriptů možnost členění skriptů do složek pokynem **Nová složka skriptů (21)**.

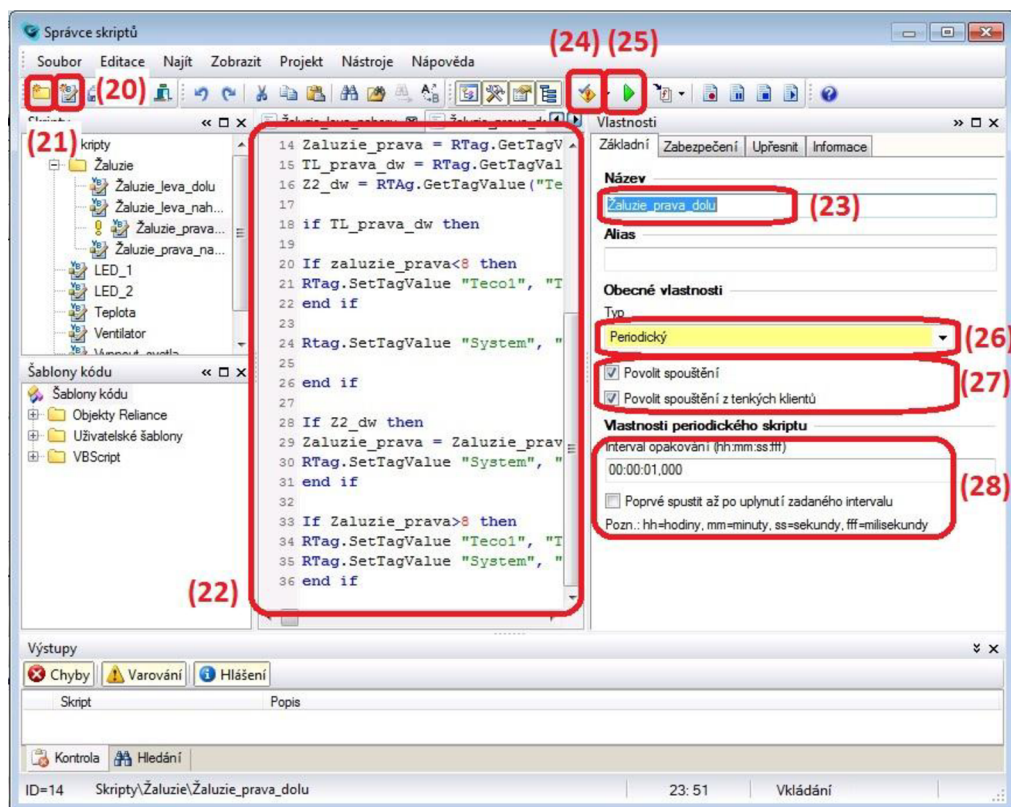
V okně (22) se píše vlastní tělo skriptu, jak už bylo uvedeno v teoretické části, k psaní skriptů slouží skriptovací jazyk VBScript. Jednotlivým skriptům můžeme přiřadit vlastní **Název (23)**. Tlačítkem **Zkontrolovat skripty (24)** můžeme ověřit, zda nemáme formální chyby ve vytvořených skriptech, kdy se nám otevře nové okno s probíhající kontrolou (Obr. 3-12). V případě, že skripty neobsahují formální chyby, je ve výstupu kontroly uvedeno

**Zjištěných chyb 0 (29).** Pokud by některý z vytvořených skriptů obsahoval formální chybu, byl by uveden počet zjištěných chyb s definováním a označením příslušných řádku.

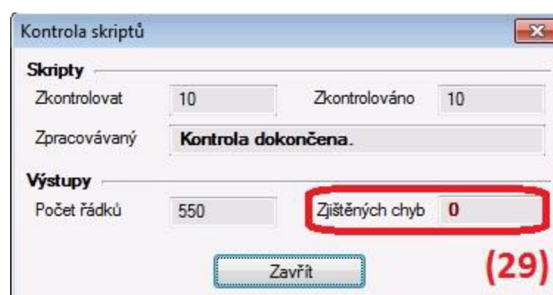
Dále dle (Obr. 3-11) můžeme spustit celý projekt tlačítkem **Spustit projekt (25)**, případně klávesou **F9** a ověřit funkce skriptů.

U jednotlivých skriptů můžeme nastavit, za jakých podmínek budou spuštěny. To provedeme volbou **Typ (26)**, kde máme na výběr z časového, událostního, klávesového, podmínkového periodického a skriptu na změnu hodnoty. U všech vyjmenovaných skriptů můžeme zatrhnout volbu **povolení spuštění a povolení spuštění z tenkých klientů (26)**.

U **Periodického skriptu (28)** (Obr. 3-11) nastavujeme periodu opakování a máme možnost zatrhnout volbu, zda první spuštění má být provedeno až po uplynutí daného časového intervalu periody, nebo první spuštění má být provedeno již při startu projektu.



Obr. 3-11 Reliance Design- Správce skriptů – 1



Obr. 3-12 Reliance Design- Správce skriptů – 2



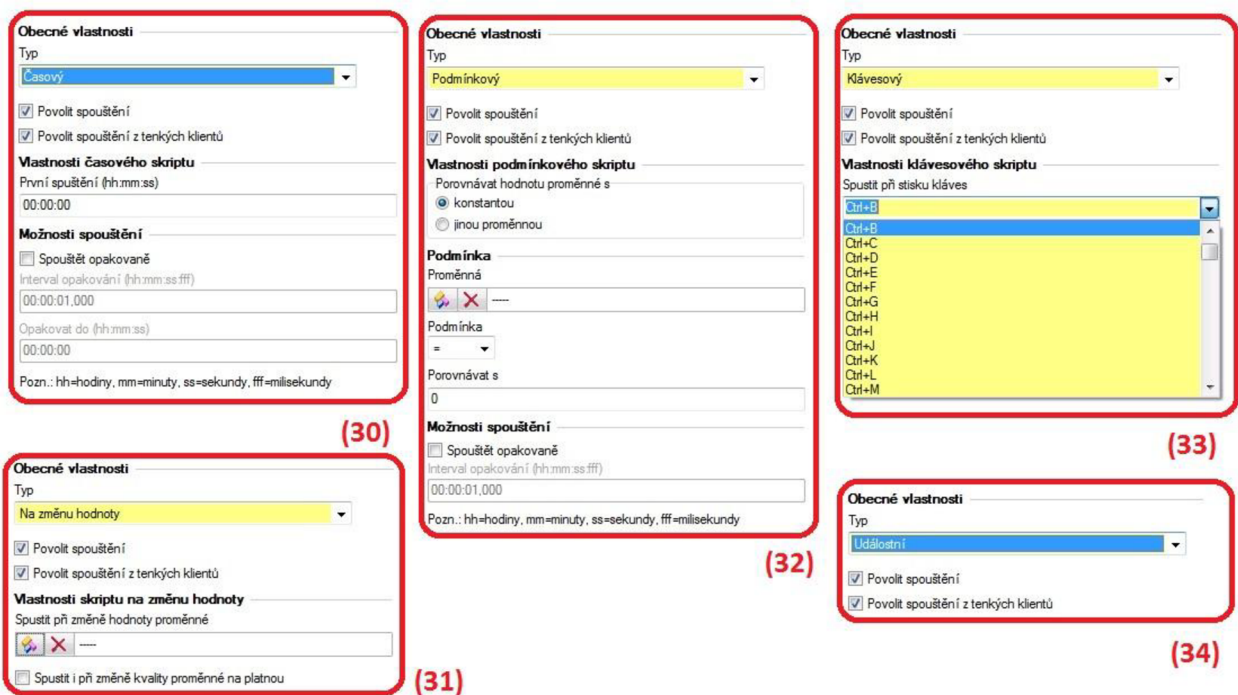
U **Časového skriptu (30)** (Obr. 3-13) nastavujeme časový údaj prvního spuštění, dle potřeby máme možnost zatrhnout volbu periodického opakování. Tato možnost nám umožní nastavit časový interval periody a současně zadat časový údaj, do kterého se má skript opakovat.

Dalším typem je skript **Na změnu hodnoty (31)**, kde můžeme zvolit příslušnou proměnnou, kterou jsme vytvořili ve správci stanic. Skript při této možnosti je spuštěn při změně její hodnoty.

**Podmínkový skript (32)** nám umožňuje spuštění v závislosti na porovnání zvolené proměnné se zvolenou konstantní hodnotou případně jinou proměnnou. Jako podmínka porovnání může být zvolen jeden ze znaků větší, menší, rovno, ne-rovno. Opět je zde možnost periodického opakování se zadaným časovým intervalem.

**Klávesový skript (33)** nám umožňuje spuštění v závislosti na stisku zvolené klávesy či kombinace kláves.

**Událostní skript (34)** umožňuje spuštění v závislosti na vzniku určitého alarmu nebo události, přiřazení skriptu dané události se provede ve správci stanic. Postup a využití je blíže popsáno zde [27].

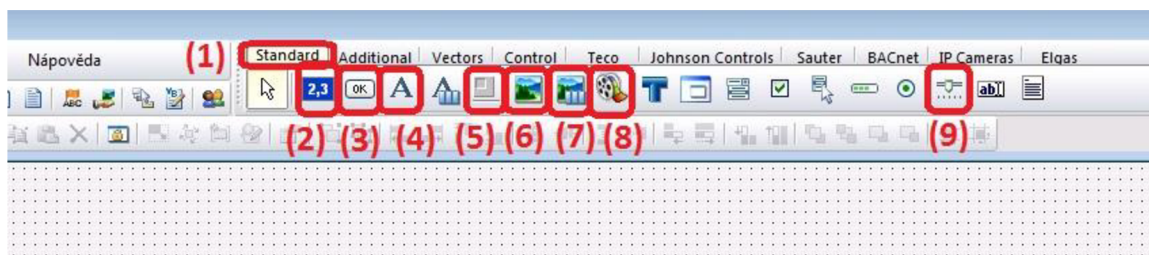


Obr. 3-13 Reliance Design- Správce skriptů – 3

### 3.2.4 Vizualizační prostředí

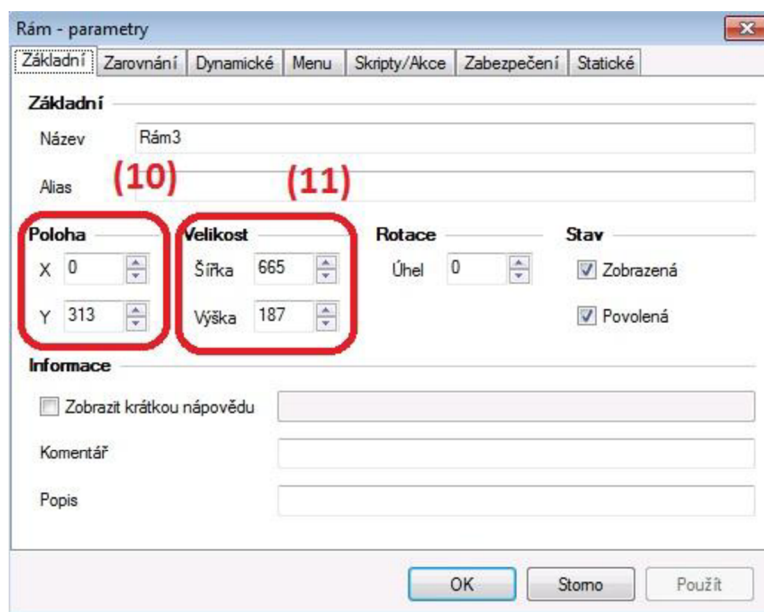
Tvorba vizualizačního prostředí probíhá vkládáním grafických a funkčních komponent. Blok s těmito komponentami je zobrazen na (Obr. 3-14) v **Hlavním menu** na jednotlivých záložkách. Na záložce **Standard (1)** jsou uvedeny nejzákladnější komponenty. Některé z nich budou níže základně popsány, pro jejich detailní popis by byl rozsah této práce nedostatečný. Vkládání komponent probíhá označením dané komponenty a následným označením místa kam ji chceme vložit. S vloženými komponentami se dá standardně manipulovat nebo upravovat jejich velikost

kurzorem myši. Vyvolání nabídky vlastností komponent probíhá dvojklikem levého tlačítka kurzoru myši.



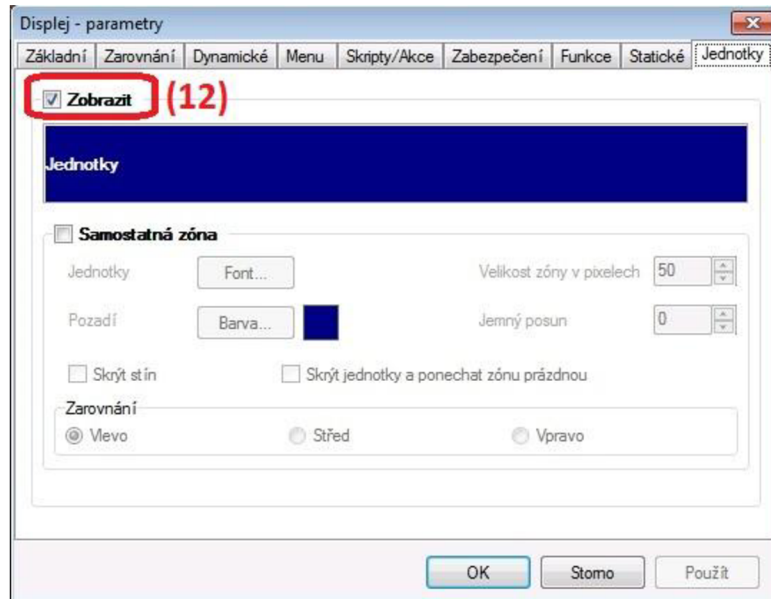
Obr. 3-14 Reliance Design- Vkládání komponent – 1

Komponentu **Rám** (5) (Obr. 3-14) můžeme využít k vytvoření grafických bloků, které nám mohou vzhledově ucelit určitou skupinu komponent, které mají společnou funkci. V případě, že chceme jednotlivé komponenty přesně vkládat do vytvořeného okna, můžeme dle (Obr. 3-15) nastavit počáteční **Polohu** (10) a **Velikost** komponenty (11). Možnost nastavení polohy a velikosti je dostupná pro řadu komponent.



Obr. 3-15 Reliance Design- Komponenta rám

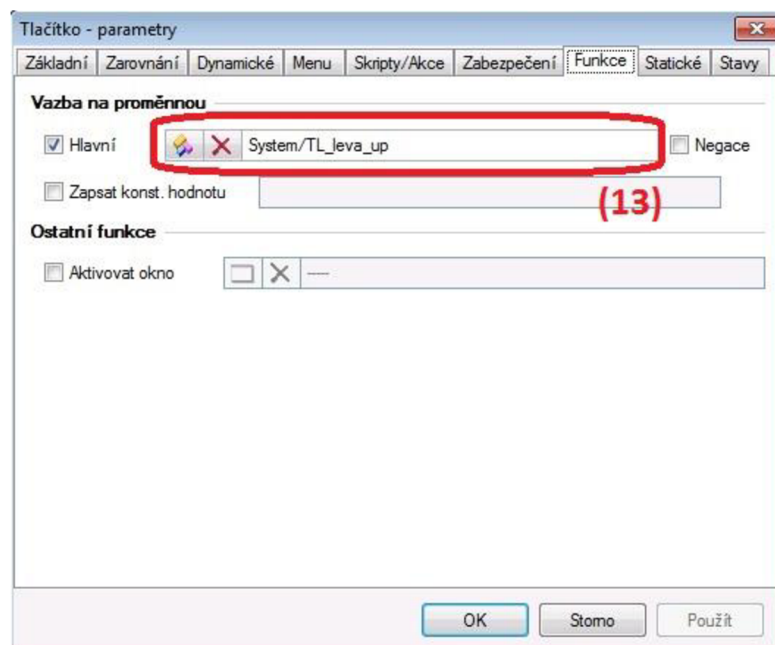
Komponentou **Displej** (2) (Obr. 3-14) můžeme zobrazit číselnou hodnotu proměnných. Dle (Obr. 3-16) můžeme na záložce jednotky zobrazit **Jednotky** (12) proměnné. Na displeji se nám zobrazí jednotka, kterou jsme definovali pro danou proměnnou ve správci stanic. Postup je popsán v (kap. 3.2.1.)



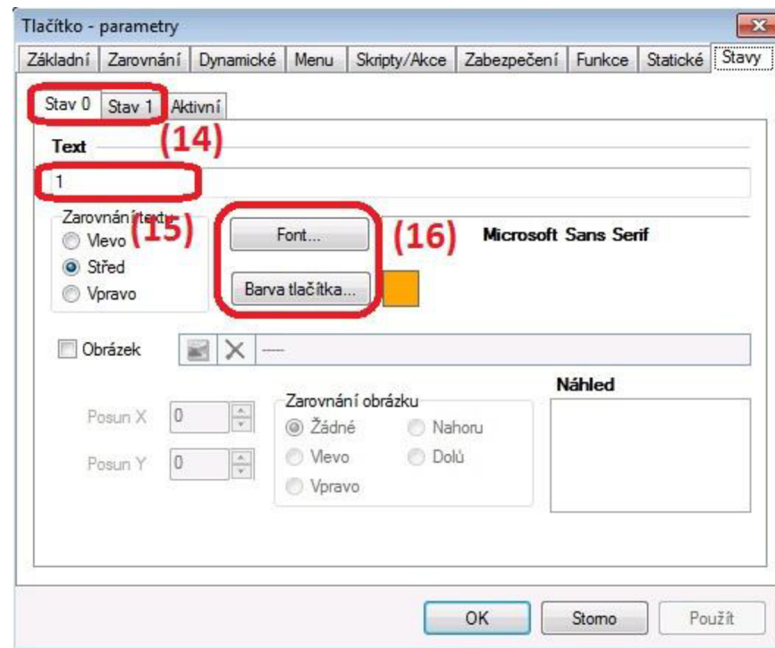
Obr. 3-16 Reliance Design- Komponenta displej

Komponentou **Tlačítko (3)** (Obr. 3-14) můžeme měnit hodnotu proměnné mezi stavy 0 a 1. Dle (Obr. 3-17) můžeme na záložce **Funkce** nastavit **Vazbu na proměnnou (13)**. Přepneme-li se na záložku **Stavy** (Obr. 3-18), pomocí tlačítek **Stav 0 a Stav 1 (14)** se můžeme přepínat mezi jednotlivými stavy a měnit jejich **Text (15)**, **Barvu tlačítka a Font písma (16)**. Text a barva tlačítka pro **Stav 0** je zobrazen, pokud není tlačítko stlačeno. V okamžiku stlačení je zobrazení dle **Stavu 1**.

Další komponentou je **Text (4)** (Obr. 3-14), kterou můžeme vkládat jednotlivé popisky do vizualizace.

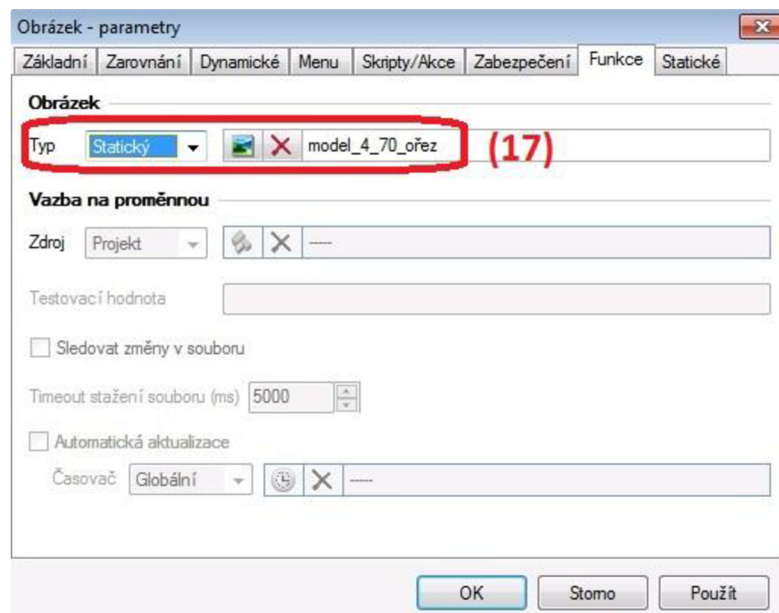


Obr. 3-17 Reliance Design- Komponenta tlačítko – 1



Obr. 3-18 Reliance Design- Komponenta tlačítko – 2

Komponentou **Obrázek** (6) (Obr. 3-14) můžeme zobrazit jednotlivé obrázky, či grafické prvky, které jsme si načetli do správce obrázků. Provedeme to dle (Obr. 3-19) **Obrázek** (17), je zde také možnost vybrat statické zobrazení, či dynamické, kdy je zobrazení obrázku vázáno na zvolenou proměnnou.



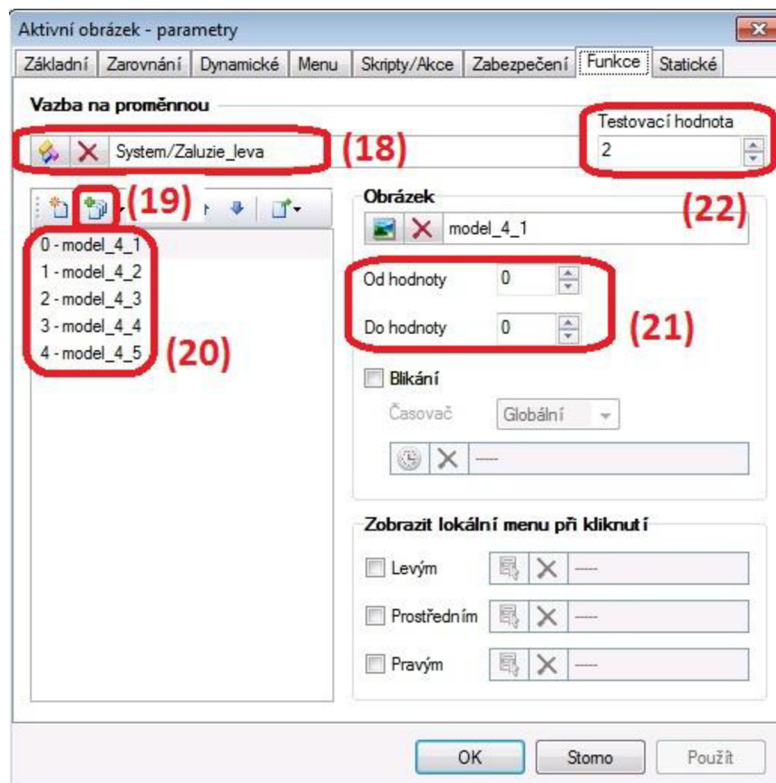
Obr. 3-19 Reliance Design – Komponenta obrázek

Obdobnou komponentou je **Aktivní obrázek** (3) (Obr. 3-14), který zobrazuje jednotlivé obrázky v závislosti na hodnotě zvolené proměnné. V nastavení, na záložce **Funkce** (Obr. 3-20), můžeme nastavit **Vazbu na proměnnou** (18), v závislosti na její změně se budou měnit zobrazované



obrázky. Jednotlivé obrázky přidáme tlačítkem **Přidat položky (19)**. Nahrané položky jsou zobrazeny v okně (20), v případě, že označíme jednotlivé řádky, můžeme pak nastavovat **počáteční a koncovou hodnotu (21)** zobrazení. Pro stav projektu, kdy není spuštěn, můžeme nastavit zobrazování daného obrázku pomocí **Testovací hodnoty (22)**.

Komponenty aktivního obrázku bylo při tvorbě vizualizační části tohoto projektu využito ve více případech. Jako příklad můžeme uvést ovládání žaluzií. Vytvořenou řídicí logikou se přiřazovala aktuální poloze žaluzií hodnota proměnné. Skrze hodnoty této proměnné jsme měli možnost zobrazit jednotlivé stavy žaluzií v grafickém prostředí. Ukázka jednotlivých stavových obrázků je uvedena na (Obr. 3-21).

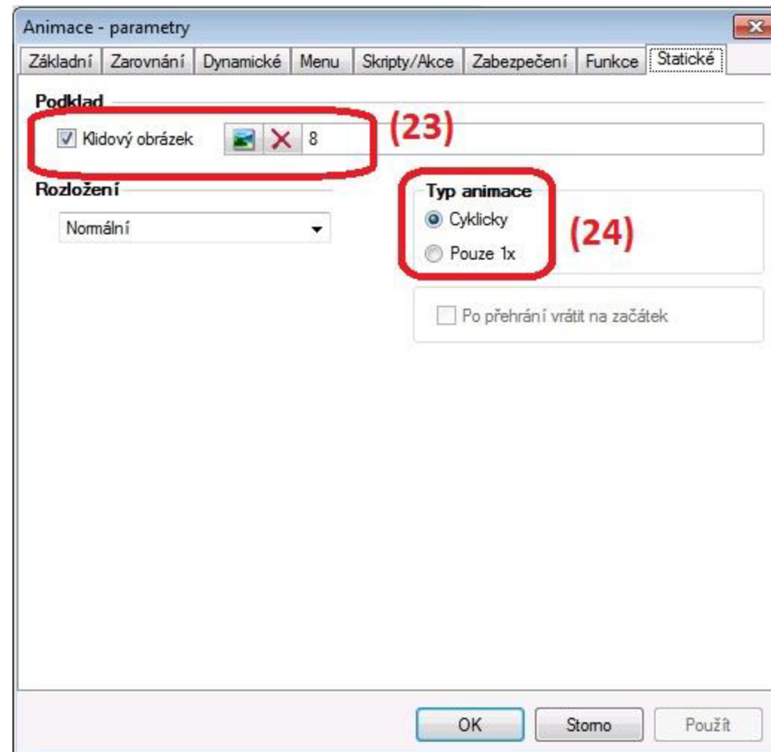


Obr. 3-20 Reliance Design – Komponenta aktivní obrázek



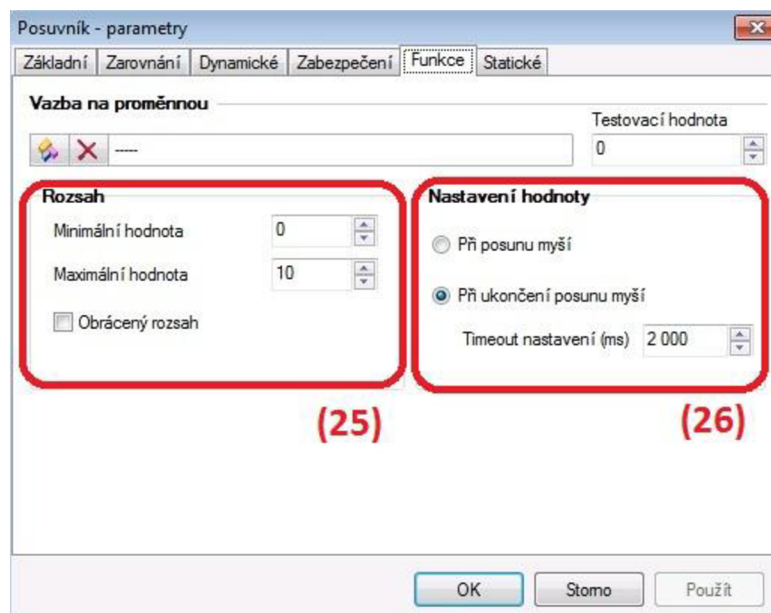
Obr. 3-21 Reliance Design – Žaluzie

Obdobou komponentou k aktivnímu obrázku je **Animace (8)** (Obr. 3-14), u které se zobrazuje klidový obrázek. V případě změny hodnoty naší zvolené proměnné se začne zobrazovat skupina obrázků cyklicky za sebou. Přidání cyklických obrázků je obdobné jako u aktivního obrázku. Klidový obrázek se přidá dle (Obr. 3-22), tlačítkem **Vybrat obrázek (23)**, případně klidový obrázek nemusí být zvolen. Dále máme na výběr mezi **cyklickým opakováním a opakováním pouze 1x (24)**.



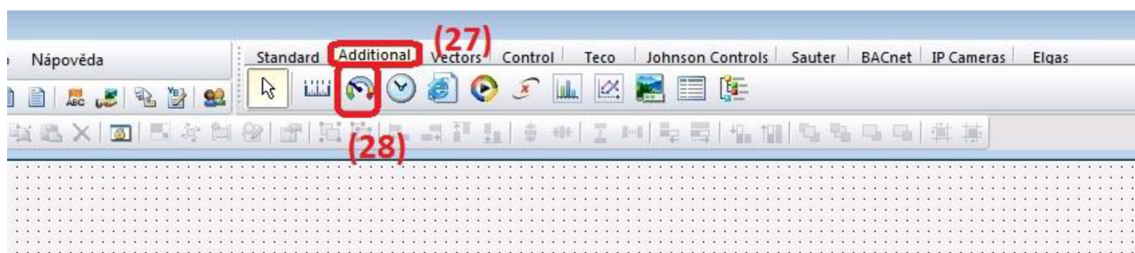
Obr. 3-22 Reliance Design- Komponenta animace

Komponentou **Posuvník (9)** (Obr. 3-14) můžeme nastavovat hodnotu proměnné, která má více jak dvě stavové hodnoty. Dle (Obr. 3-23) můžeme nastavit **Minimální a maximální hodnotu (25)** zvolené proměnné, případně nastavit obrácený rozsah. **Nastavení hodnoty (26)** si můžeme zvolit při posunutí myši nebo se zadaným zpožděním.

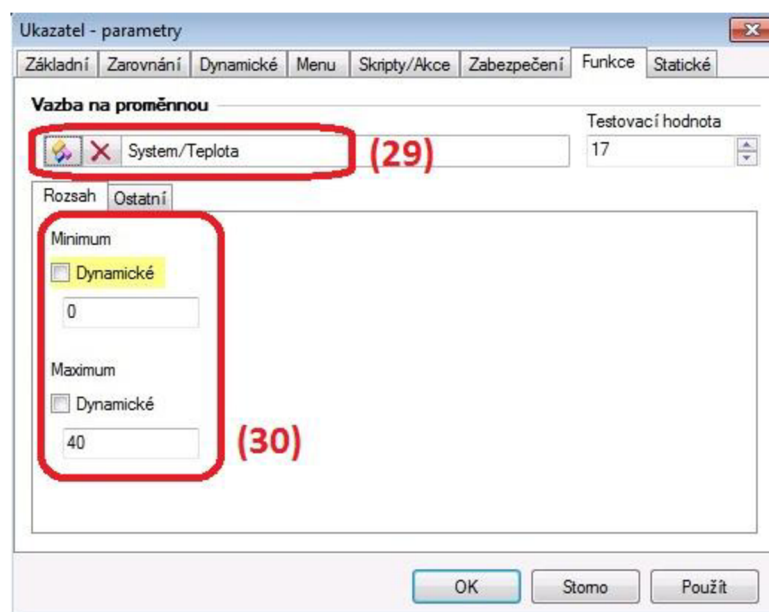


Obr. 3-23 Reliance Design- Komponenta posuvník

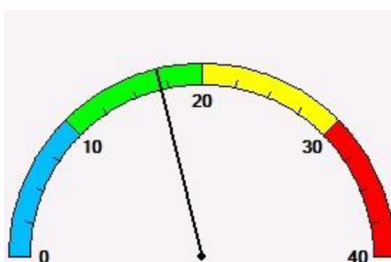
Další komponenty zobrazíme dle (Obr. 3-24) na záložce **Další (27)**, například komponenta **Ukazatel (28)**. Komponentou ukazatel lze zobrazit hodnoty proměnné na obloukové stupnici. Dle (Obr. 3-25) nastavíme **Vazbu na proměnnou (29)**. Dále **Minimum a Maximum (30)**, které lze nastavit jako dynamické na zadané proměnné. Dle (Obr. 3-27) přepneme na záložku **Statické / Meze (31)**, v této záložce můžeme zatrhnout **využití jednotlivých mezí (32)** a vybrat barvu mezí. Pokud jsme na ukazatel navázali proměnnou, u které jsme definovali hodnoty mezí, jak bylo popsáno v (kap. 3.2.1), pak se nám tyto meze a jejich příslušné barvy zobrazí na ukazateli jako na (Obr. 3-26).



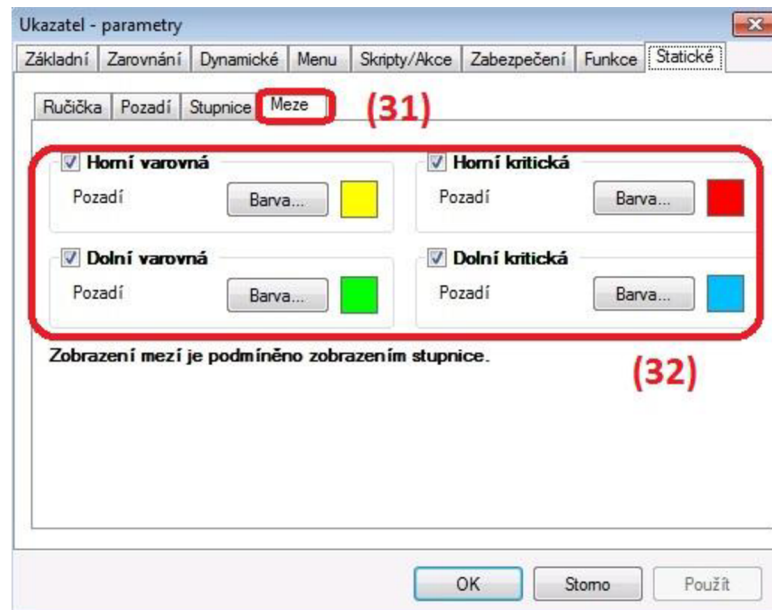
Obr. 3-24 Reliance Design- Vkládání komponent – 2



Obr. 3-25 Reliance Design- Komponenta ukazatel – 1



Obr. 3-26 Reliance Design- Komponenta ukazatel – 2



Obr. 3-27 Reliance Design- Komponenta ukazatel – 3

### 3.2.5 Řídící logika

Jak již bylo uvedeno výše, k řízení procesů je nutné vytvořit řídicí logiku. V systému Reliance slouží k této činnosti správce skriptů, jehož funkce je popsána v (kap. 3.2.3).

V následujících podkapitolách jsou popsány jednotlivé skripty, které byly v projektu vytvořeny. U všech skriptů je spuštění nastaveno periodicky, s intervalem opakování jedné sekundy.

#### 3.2.5.1 Skript ovládání žaluzií směrem dolů

U ovládání žaluzií je nutné jednotlivým polohám žaluzií přiřazovat proměnnou, na základě které se zobrazuje stav zavření a otevření jednotlivé žaluzie, viz. (kap. 3.2.4) – komponenta aktivní obrázek. Pro každou žaluzii je vytvořen samostatný skript směru ovládání dolů a nahoru. Celkem jsou tedy pro ovládání obou žaluzií vytvořeny čtyři skripty.

V úvodní části skriptu ovládání žaluzií směrem dolů jsou definovány proměnné. Následuje podmínka na stlačení vizualizačního tlačítka žaluzie směrem dolů. Pokud je tlačítko stlačeno, následuje další podmínka, která ověřuje, zda je žaluzie v jiné než krajní dolní poloze. Pokud žaluzie není stažena v krajní dolní poloze, pak je aktivována proměnná ze stanice TECO1, která spustí chod žaluzie směrem dolů. Následně je proměnné vizualizačního tlačítka nastavena nulová hodnota, aby se chovalo jako tlačítko a ne spínač. Původní podmínka na stlačení vizualizačního tlačítka je ukončena. Následuje podmínka, která sleduje kontrolku spuštění žaluzie ze stanice TECO1. V případě, že žaluzie je v chodu směrem dolů, je proměnné, která sleduje polohu žaluzií přičtena hodnota 1. Celkový interval chodu žaluzií z jedné krajní polohy do druhé je nastaven na 8 sekund. Po přičtení hodnoty 1 je podmínka ukončena. Následuje další podmínka, která sleduje, zda žaluzie není v krajní spodní poloze. V případě, že proměnná, která sleduje polohu žaluzií, nabude hodnotu větší jak 8, je proměnné nastavena hodnota 8 a deaktivována proměnná stanice

TECO1, která vypne chod žaluzií směrem dolů. Jak už bylo uvedeno výše, skript je periodický s dobou periody jedné sekundy, a tedy přičítané hodnotě 1 odpovídá jedna sekunda.

```
Option Explicit

dim TL1a_dw, Zaluzie_leva, TL_leva_dw, Z1_dw

TL1a_dw = RTag.GetTagValue("Teco1", "TL1a_dw")
Zaluzie_leva = RTag.GetTagValue("System", "Zaluzie_leva")
TL_leva_dw = RTag.GetTagValue("System", "TL_leva_dw")
Z1_dw = RTag.GetTagValue("Teco1", "Z1_dw")

If TL_leva_dw then

If zaluzie_leva<8 then
RTag.SetTagValue "Teco1", "TL1a_dw", 1
end if

Rtag.SetTagValue "System", "TL_leva_dw", 0

end if

If Z1_dw then
Zaluzie_leva = Zaluzie_leva + 1
RTag.SetTagValue "System", "Zaluzie_leva", Zaluzie_leva
end if

If Zaluzie_leva>8 then
RTag.SetTagValue "Teco1", "TL1a_dw", 1
RTag.SetTagValue "System", "Zaluzie_leva", 8
end if
```

### 3.2.5.2 Skript ovládání žaluzií směrem nahoru

Skript pro ovládání žaluzií směrem nahoru je velmi obdobný, jako předchozí uvedený – směrem dolů. V úvodní části skriptu ovládání žaluzií směrem nahoru jsou opět definovány proměnné. Následuje podmínka na stlačení vizualizačního tlačítka žaluzie směrem nahoru. Pokud je tlačítko stlačeno, následuje další podmínka, která ověřuje, zda je žaluzie v jiné než krajní horní poloze. Pokud žaluzie není vytažena v krajní horní poloze, pak je aktivována proměnná ze stanice TECO1, která spustí chod žaluzie směrem nahoru. Následně je proměnné vizualizačního tlačítka nastavena nulová hodnota, aby se chovalo jako tlačítko a ne spínač. Původní podmínka na stlačení vizualizačního tlačítka je ukončena. Následuje podmínka, která sleduje kontrolku spuštění žaluzie ze stanice TECO1. V případě, že žaluzie je v chodu směrem nahoru, je proměnné, která sleduje polohu žaluzií odečtena hodnota 1 a následně podmínka ukončena. Následuje další podmínka, která sleduje, zda žaluzie není v krajní horní poloze. V případě, že proměnná, která sleduje polohu žaluzií, nabude hodnotu menší jak 0, je proměnné nastavena hodnota 0 a deaktivována proměnná stanice TECO1, která vypne chod žaluzií směrem nahoru.

```
Option Explicit

dim TL1a_up, Zaluzie_leva, TL_leva_up, Z1_up
```

```
TL1a_up = RTag.GetTagValue("Teco1", "TL1a_up")
Zaluzie_leva = RTag.GetTagValue("System", "Zaluzie_leva")
TL_leva_up = RTag.GetTagValue("System", "TL_leva_up")
Z1_up = RTag.GetTagValue("Teco1", "Z1_up")

If TL_leva_up then

If zaluzie_leva>0 then
RTag.SetTagValue "Teco1", "TL1a_up", 1
end if

Rtag.SetTagValue "System", "TL_leva_up", 0

end if

If Z1_up then
Zaluzie_leva = Zaluzie_leva - 1
RTag.SetTagValue "System", "Zaluzie_leva", Zaluzie_leva
end if

If Zaluzie_leva<0 then
RTag.SetTagValue "Teco1", "TL1a_dw", 1
RTag.SetTagValue "System", "Zaluzie_leva", 0
end if
```

### 3.2.5.3 Skript ovládání LED pásků

Panel Foxtrot obsahuje dva kusy LED pásků, které jsou ovládány modulem C-DM-0006M ULED. Modul je aktor s šesti nezávislými kanály pro proporcionální řízení svitu LED pásků, se společnou anodou. Jednotlivé kanály jsou samostatně adresovatelné a ovladatelné v rozsahu 0 - 100 % napájecího napětí 12 nebo 24 V[28].

Pro jejich ovládání jsou vytvořeny dva skripty, které jsou rozdílné jen v nastavených proměnných. Pro vizualizační prostředí jednoho pásku je vytvořeno osm tlačítek, kterými se nastavují jednotlivé barvy. Ovládání je doplněno devátým tlačítkem, kterým se LED pásek vypne, resp. nastaví nulové úrovně napájecího napětí.

Opět se jedná o periodické skripty s periodou opakování jedné sekundy, kdy jsou v úvodu definovány proměnné. Skript pokračuje podmínkou na stlačení vizualizačního tlačítka dané barvy, např. červené. Pokud je tlačítko stlačeno, podmínka je splněna a následně jsou nastaveny procentní úrovně napájecího napětí RGB kanálů tak, aby byla zobrazena daná barva. Jelikož nastavujeme procentní úrovně napájecího napětí, je nutné přepočítat barevnou skladbu z rozsahů RGB 0 – 255 na rozsah 0 – 100 % napájecího napětí. Červené barvě (R=255, G=0, B=0) odpovídají procentní hodnoty napájecího napětí pro jednotlivé kanály (R=100, G=0, B=0).

Dále podmínka pokračuje nastavením proměnné vizualizačního tlačítka na nulovou hodnotu, aby se chovalo jako tlačítko a ne spínač. I u vizualizačního zobrazení LED diod je využito funkce aktivního obrázku, viz. (kap. 3.2.4). Zobrazení je provedeno v podobě vytvořené grafiky LED pásků (Obr. 3-28), kdy je změněna barva grafického prvku dle aktuálně nastavené barvy. Pro tento aktivní obrázek je nutné v podmínce nastavení dané barvy přiřadit hodnotu stanovené proměnné, podle níž se zobrazují odpovídající grafické prvky. Po přiřazení hodnoty proměnné je podmínka ukončena. Následují obdobné podmínky pro zbývající tlačítka barev. Jako poslední je



podmínka pro deváté tlačítko, u kterého, jak už bylo uvedeno výše, se nastaví nulové úrovně napájecího napětí.



Obr. 3-28 Reliance Design- grafické prvky LED pásků

Option Explicit

```
dim cervena_1, oranzoava_1, zluta_1, zelena_1, zelenomodra_1, modra_1,
fialova_1, bila_1, cerna_1, LED1, LED2, LED3, LED1_pocitadlo
```

```
cervena_1 = RTag.GetTagValue("System", "Červená_1")
oranzoava_1 = RTag.GetTagValue("System", "Oranžová_1")
zluta_1 = RTag.GetTagValue("System", "Žlutá_1")
zelena_1 = RTag.GetTagValue("System", "Zelená_1")
zelenomodra_1 = RTag.GetTagValue("System", "Zelenomodrá_1")
modra_1 = RTag.GetTagValue("System", "Modrá_1")
fialova_1 = RTag.GetTagValue("System", "Fialová_1")
bila_1 = RTag.GetTagValue("System", "Bílá_1")
cerna_1 = RTag.GetTagValue("System", "Černá_1")
LED1_pocitadlo = RTag.GetTagValue("System", "LED1_pocitadlo")
```

```
LED1 = RTag.GetTagValue("Teco1", "LED1")
LED2 = RTag.GetTagValue("Teco1", "LED2")
LED3 = RTag.GetTagValue("Teco1", "LED3")
```

```
If cervena_1 then
RTag.SetTagValue "Teco1", "LED1", 100
RTag.SetTagValue "Teco1", "LED2", 0
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "System", "Červená_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 1
end if
```

```
If oranzoava_1 then
RTag.SetTagValue "Teco1", "LED1", 100
RTag.SetTagValue "Teco1", "LED2", 65
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "System", "Oranžová_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 2
end if
```

```
If zluta_1 then
RTag.SetTagValue "Teco1", "LED1", 100
RTag.SetTagValue "Teco1", "LED2", 100
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "System", "Žlutá_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 3
end if
```

```
If zelena_1 then
RTag.SetTagValue "Teco1", "LED1", 0
RTag.SetTagValue "Teco1", "LED2", 100
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "System", "Zelená_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 4
end if

If zelenomodra_1 then
RTag.SetTagValue "Teco1", "LED1", 0
RTag.SetTagValue "Teco1", "LED2", 100
RTag.SetTagValue "Teco1", "LED3", 100
RTag.SetTagValue "System", "Zelenomodrá_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 5
end if

If modra_1 then
RTag.SetTagValue "Teco1", "LED1", 0
RTag.SetTagValue "Teco1", "LED2", 0
RTag.SetTagValue "Teco1", "LED3", 100
RTag.SetTagValue "System", "Modrá_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 6
end if

If fialova_1 then
RTag.SetTagValue "Teco1", "LED1", 100
RTag.SetTagValue "Teco1", "LED2", 0
RTag.SetTagValue "Teco1", "LED3", 100
RTag.SetTagValue "System", "Fialová_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 7
end if

If bila_1 then
RTag.SetTagValue "Teco1", "LED1", 100
RTag.SetTagValue "Teco1", "LED2", 100
RTag.SetTagValue "Teco1", "LED3", 100
RTag.SetTagValue "System", "Bílá_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 8
end if

If cerna_1 then
RTag.SetTagValue "Teco1", "LED1", 0
RTag.SetTagValue "Teco1", "LED2", 0
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "System", "Černá_1", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 0
end if
```

### 3.2.5.4 Skript Teplota

Skript teplota slouží k simulování teploty, v závislosti na stavu zapnutého topení nebo chlazení. Opět se jedná o periodický skript s periodou jedné sekundy. Po úvodním načtení proměnných následují podmínky. Následují podmínky, které sledují kontrolky spuštění topení a spuštění chlazení ze stanice TECO1. V případě, že je spuštěno Topení, pak je fiktivní proměnné přičtena hodnota 1. V případě chlazení je odečtena hodnota 1. Po těchto podmínkách následují



další podmínky, které omezují maximální nárůst, či pokles teploty. První podmínka sleduje maximální hodnotu teploty. V případě, že teplota bude mít hodnotu proměnné vyšší jak 40, bude deaktivována proměnná stanice TECO1, která vypne chod topení. Druhá podmínka sleduje minimální hodnotu teploty. V případě, že teplota bude mít hodnotu proměnné nižší jak 0, bude deaktivována proměnná stanice TECO1, která vypne chod chlazení. Následně je hodnota teploty ze skriptu zapsána do stanice System.

```
Option Explicit

dim Topeni, Chlazení, Teplota

Teplota = RTag.GetTagValue("System", "Teplota")
Topeni = RTag.GetTagValue("Teco1", "Topeni")
Chlazení = RTag.GetTagValue("Teco1", "Chlazení")

If Topeni then
    Teplota = Teplota + 1
end if

If Chlazení then
    Teplota = Teplota - 1
end if

If Teplota => 41 then
    RTag.SetTagValue "Teco1", "TL2a_up", 1
    Teplota = 40
end if

If Teplota =< -1 then
    RTag.SetTagValue "Teco1", "TL2a_dw", 1
    Teplota = 0
end if

RTag.SetTagValue "System", "Teplota", Teplota
```

### 3.2.5.5 Skript ventilátor

Animace ventilátoru, (viz. *kap. 3.2.4*) – komponenta animace, má být spuštěna v případě, kdy je zapnuto topení nebo chlazení. Tato animace vytváří pomocí série obrázků dojem, že se ventilátor otáčí. V případě, že není zapnuto ani topení, ani chlazení, má být ventilátor v klidovém stavu se statickým obrázkem. Abychom navázali animaci na obě proměnné, byl vytvořen následující skript.

Skript je opět periodický s periodou opakování jedna sekunda. V úvodu jsou načteny proměnné a následují dvě vnořené podmínky, které sledují spuštění topení a chlazení ze stanice TECO1. V případě, že je topení nebo chlazení spuštěno, je fiktivní proměnné ventilátor ze stanice System přiřazena hodnota 1, v opačném případě 0. Následně je hodnota proměnné ventilátor ze skriptu zapsána do stanice System.

Zmíněná fiktivní proměnná je napojena na komponentu animace ventilátoru.

```
dim Topeni, Chlazení, Ventilator

Topeni = RTag.GetTagValue("Teco1", "Topeni")
Chlazení = RTag.GetTagValue("Teco1", "Chlazení")
Ventilator = RTag.GetTagValue("System", "Ventilator")

if Topeni then
    Ventilator = 1
else
    if Chlazení then
        Ventilator = 1
    else
        Ventilator = 0
    end if
end if

RTag.SetTagValue "System", "Ventilator", Ventilator
```

### 3.2.5.6 Skript vypnutí všech prvků

Následující skript, slouží k vypnutí všech prvků, které jsou osazeny na panelu. Opět se jedná o periodický skript s periodou opakování jedné sekundy. V úvodu jsou načteny proměnné. Následuje podmínka na stlačení vizualizačního tlačítka, kterým mají být vypnuty všechny prvky. Do této podmínky jsou vnořeny další podmínky. Pokud je tlačítko stlačeno, je podmínka splněna a následně jsou vypínány postupně svítidla, LED pásy, žaluzie, topení, chlazení. U většiny prvků je vložena podmínka na jejich kontrolku a následně aktivováno jejich příslušné tlačítko. LED páskům jsou přímo bez podmínky nastaveny nulové úrovně napájecího napětí. Současně zde musíme opět nastavit hodnotu proměnných, odpovídajících vypnutému stavu LED pásků, kterou se řídí zobrazení grafiky LED pásků na vizualizaci, jak bylo popsáno v (kap. 3.2.5.3). Podmínka na stlačení vizualizačního tlačítka, kterým měli být vypnuty všechny prvky je ukončena, a následuje nastavení proměnné vizualizačního tlačítka na nulovou hodnotu, aby se chovalo jako tlačítko a ne spínač.

```
dim Vypnout_vse, TL1a_dw, TL1a_up, TL1b_dw, TL1b_up, TL2a_dw, TL2a_up,
Topeni, Chlazení, Z1_dw, Z1_up, Z2_dw, Z2_up, TL2b_dw, TL2b_up, SP1, SP3,
LED1, LED2, LED3, LED4, LED5, LED6, LED1_pocitadlo, LED2_pocitadlo

Vypnout_vse = RTag.GetTagValue("System", "Vypnout_vse")
TL1a_dw = RTag.GetTagValue("Teco1", "TL1a_dw")
TL1a_up = RTag.GetTagValue("Teco1", "TL1a_up")
TL1b_dw = RTag.GetTagValue("Teco1", "TL1b_dw")
TL1b_up = RTag.GetTagValue("Teco1", "TL1b_up")
TL2a_dw = RTag.GetTagValue("Teco1", "TL2a_dw")
TL2a_up = RTag.GetTagValue("Teco1", "TL2a_up")
Topeni = RTag.GetTagValue("Teco1", "Topeni")
Chlazení = RTag.GetTagValue("Teco1", "Chlazení")
Z1_dw = RTag.GetTagValue("Teco1", "Z1_dw")
Z1_up = RTag.GetTagValue("Teco1", "Z1_up")
Z2_dw = RTag.GetTagValue("Teco1", "Z2_dw")
Z2_up = RTag.GetTagValue("Teco1", "Z2_up")
TL2b_dw = RTag.GetTagValue("Teco1", "TL2b_dw")
TL2b_up = RTag.GetTagValue("Teco1", "TL2b_up")
SP1 = RTag.GetTagValue("Teco1", "SP1")
```

```
SP3 = RTag.GetTagValue("Teco1", "SP3")
LED1 = RTag.GetTagValue("Teco1", "LED1")
LED2 = RTag.GetTagValue("Teco1", "LED2")
LED3 = RTag.GetTagValue("Teco1", "LED3")
LED4 = RTag.GetTagValue("Teco1", "LED4")
LED5 = RTag.GetTagValue("Teco1", "LED5")
LED6 = RTag.GetTagValue("Teco1", "LED6")
LED1_pocitadlo = RTag.GetTagValue("System", "LED1_pocitadlo")
LED2_pocitadlo = RTag.GetTagValue("System", "LED2_pocitadlo")

If Vypnout_vse then

If SP1 then
RTag.SetTagValue "Teco1", "TL2b_up", 1
end if

If SP3 then
RTag.SetTagValue "Teco1", "TL2b_dw", 1
end if

RTag.SetTagValue "Teco1", "LED1", 0
RTag.SetTagValue "Teco1", "LED2", 0
RTag.SetTagValue "Teco1", "LED3", 0
RTag.SetTagValue "Teco1", "LED4", 0
RTag.SetTagValue "Teco1", "LED5", 0
RTag.SetTagValue "Teco1", "LED6", 0
RTag.SetTagValue "System", "LED1_pocitadlo", 0
RTag.SetTagValue "System", "LED2_pocitadlo", 0

If Z1_up then
RTag.SetTagValue "Teco1", "TL1a_up", 1
end if

If Z1_dw then
RTag.SetTagValue "Teco1", "TL1a_dw", 1
end if

If Z2_up then
RTag.SetTagValue "Teco1", "TL1b_up", 1
end if

If Z2_dw then
RTag.SetTagValue "Teco1", "TL1b_dw", 1
end if

If Topeni then
RTag.SetTagValue "Teco1", "TL2a_up", 1
end if

If Chlazení then
RTag.SetTagValue "Teco1", "TL2a_dw", 1
end if

end if

Rtag.SetTagValue "System", "Vypnout_vse", 0
```

### 3.2.5.7 Skript vypnutí osvětlení

Skript na vypnutí osvětlení je velmi obdobný jako skript na vypnutí všech prvků. V podstatě se jedná o jeho dílčí část, kdy jsou vypínány jen prvky osvětlení, tedy svítidla a LED pásy. Jelikož se jedná o dílčí část předchozího skriptu, není zde skript uveden.

## 3.3 Řídící vizualizační prostředí – popis aplikace

K zadanému panelu Foxtrot (*Obr. 3-1*) bylo vytvořeno řídicí vizualizační prostředí (*Obr. 3-29*) pro ovládání prvků osazených na panelu. Jsou zde ovládány dva světelné okruhy, dva LED pásy, pohyb žaluzií, topení a chlazení.

Světelné okruhy jsou spínány každý samostatně, prvním stlačením tlačítka příslušného okruhu je okruh spuštěn, což znázorní „rozsvícené“ žárovky ve vizualizačním prostředí. Dalším stlačením je okruh vypnut.

Chlazení opět spustíme prvním stlačením tlačítka, druhým stlačením tlačítka vypneme. Chlazení může být vypnuto i stlačením tlačítka topení. V případě, kdy je chlazení v chodu klesá fiktivní teplota každou sekundu o 1 °C. Aktuální hodnota teploty je zobrazena ve vizualizačním prostředí na displeji a ručičkovém ukazateli. Teplota klesá až do minimální hodnoty 0 °C, následně je chlazení vypnuto.

Ovládání spouštění topení je obdobné jako u chlazení. U spuštěného topení roste teplota o 1 °C za každou sekundu, maximální teplota je nastavena na 40 °C, v případě dosažení této teploty je topení vypnuto.

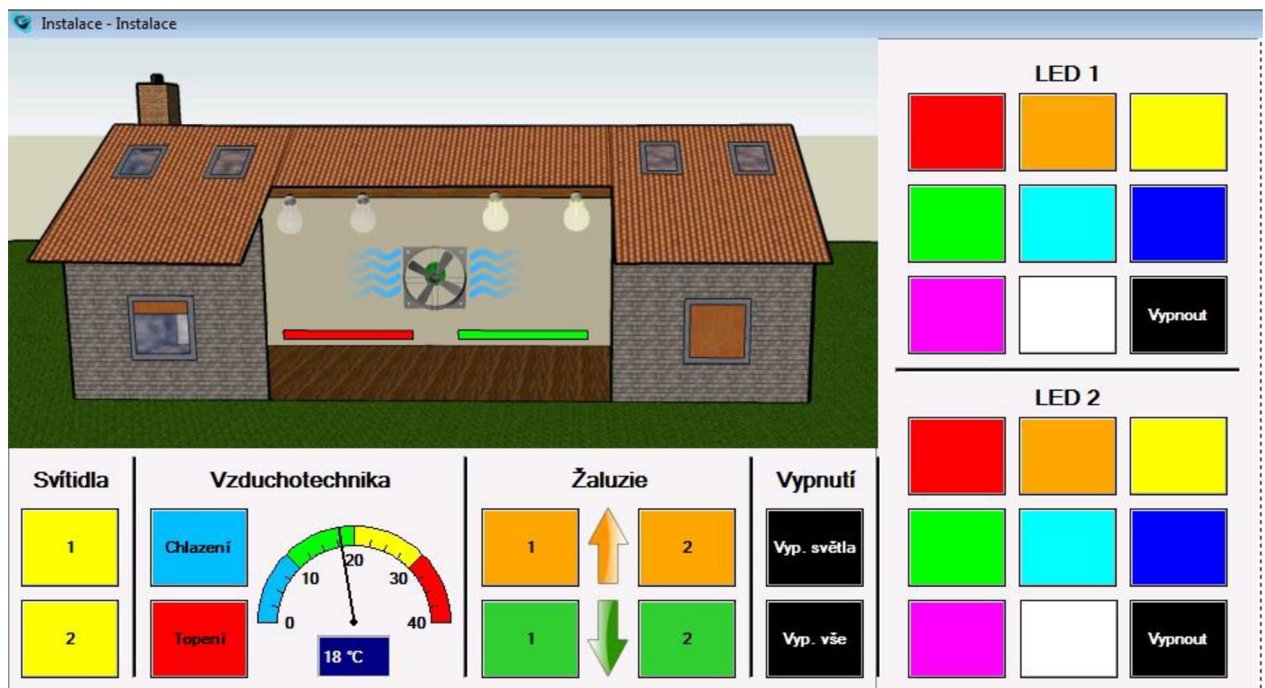
Ve vizualizačním prostředí je k signalizaci chodu chlazení či topení využito funkce ventilátoru. Pokud není spuštěno chlazení ani topení, pak je ventilátor vypnut. V případě, že je spuštěno chlazení nebo topení začne se ventilátor otáčet a okolo ventilátoru začne proudit v případě chlazení modrý vzduch, v případě topení červený.

Další prvky osazené na panelu jsou dvě žaluzie. Každá je ovládána samostatně ve směru nahoru a dolů. Celkem jsou k jejich ovládání využita čtyři tlačítka. Aktuální poloha žaluzií je zobrazena ve vizualizačním prostředí. Pokud je žaluzie v krajní horní poloze, pak tlačítko, které ovládá pohyb směrem dolů, nereaguje. Opačně, pokud je žaluzie v krajní dolní poloze, pak tlačítko, které ovládá pohyb směrem nahoru, nereaguje. Pohyb žaluzií lze zastavit druhým stlačením tlačítka, případně stlačením tlačítka protisměru. Celkový cyklus zavření je osm sekund.

Na panelu jsou také osazené dva LED pásy. Každý pásek je ovládán samostatně pomocí devíti tlačítek. Osmi tlačítky si uživatel může rozsvítit jednotlivé přednastavené barvy. Barvy můžeme mezi sebou přepínat stlačením příslušného tlačítka s odpovídající barvou. K vypnutí LED pásky slouží deváté ovládací tlačítko. Ve vizualizačním prostředí je k signalizaci chodu LED pásek využito grafických prvků, které mají znázorňovat LED pásy. Grafické prvky jsou zbarveny podle navolených barev. V případě vypnutí LED pásek jsou grafické prvky zbarveny do šedé barvy.

Řídící panel je doplněn o dvě vypínací tlačítka. První, které vypne světla, tedy oba světelné okruhy a LED pásy. Druhé tlačítko vypne veškeré prvky, které jsou v danou chvíli aktivní.

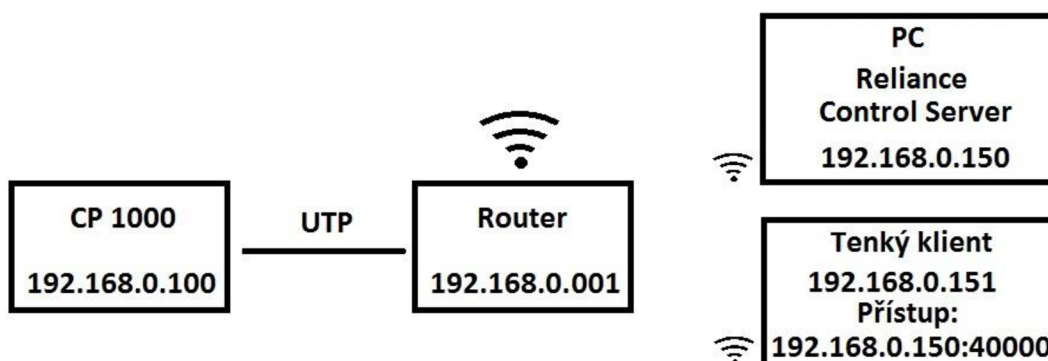
Aktuální stav dle (*Obr. 3-29*), spuštěn světelný okruh 2, spuštěno chlazení, aktuální teplota 18 °C, žaluzie 1 je z části zavřena, žaluzie 2 je v krajní dolní poloze, LED1 je nastavena na červenou barvu a LED2 na zelenou barvu.



Obr. 3-29 Reliance Design- vizualizační prostředí

## 4 VZDÁLENÉ ŘÍZENÍ ELEKTROINSTALACE V REÁLNÉM ČASE

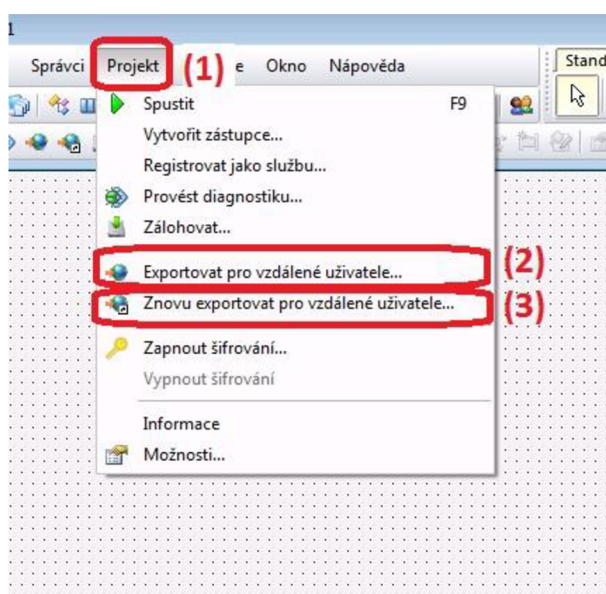
System Reliance umožňuje vzdálené řízení v reálném čase s možností vizualizace na tenkých klientech pomocí modulu Control Server. Aby bylo možné vytvořit vzdálené řízení, je nutné vytvořit datovou komunikaci. K této komunikaci byl využit router, který je osazený na panelu Foxtrot. Schéma datové komunikace je uvedena na (Obr. 4-1).



Obr. 4-1 Datová komunikace

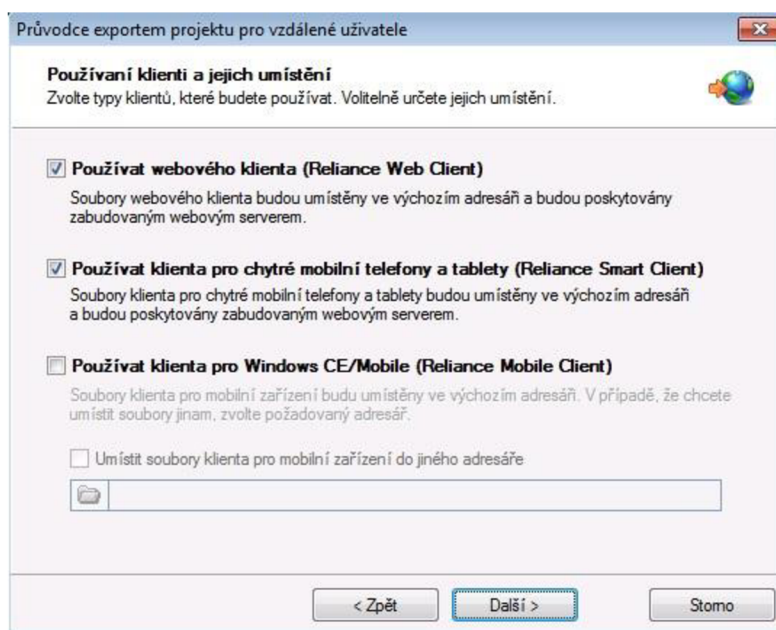
### 4.1 Export pro vzdálené uživatele v systému Reliance

Pro tvorbu vzdáleného řízení je nutné z modulu Reliance Design exportovat projekt pro vzdálené uživatele. Export provedeme následovně dle (Obr. 4-2), volbou **Hlavní menu / Projekt (1) / Exportovat pro vzdálené uživatele (2)**.

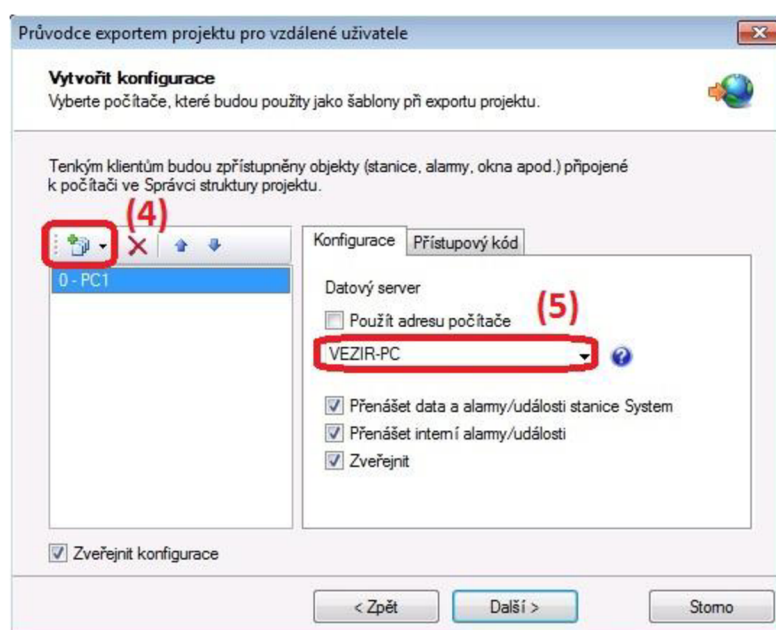


Obr. 4-2 Export pro vzdálené uživatele – 1

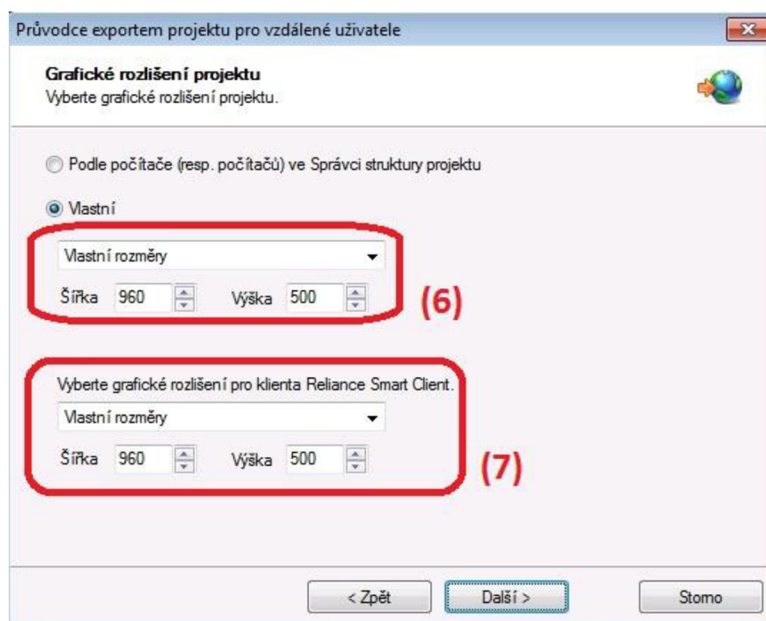
Po této volbě se nám zobrazí nové okno s informacemi týkajícími se exportu. Na další stránku pokračujeme tlačítkem **Další**. Následující strana (Obr. 4-3), dává na výběr, pro který typ uživatelů si přejeme projekt exportovat. Jednotliví klienti jsou popsáni v (kap. 2.1.2.3). Po zvolení požadovaných klientů pokračujeme tlačítkem **Další**. Na následující straně (Obr. 4-4), vybíráme počítače, které budou použity jako šablony exportu. Jednotlivé PC přidáváme pomocí volby **Přidat položky (4)**. Pokud označíme příslušnou položku, můžeme v **okně (5)** zvolit název PC, případně IP adresu.



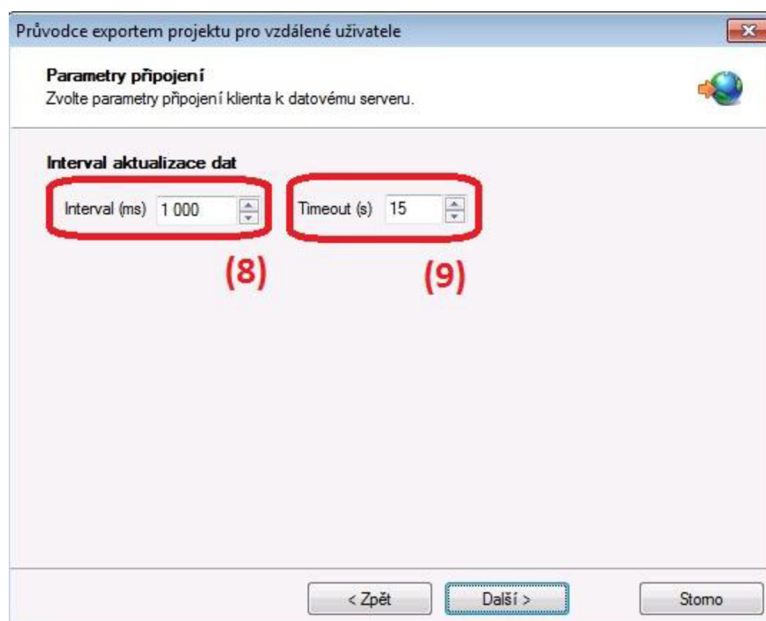
Obr. 4-3 Export pro vzdálené uživatele – 2



Obr. 4-4 Export pro vzdálené uživatele – 3



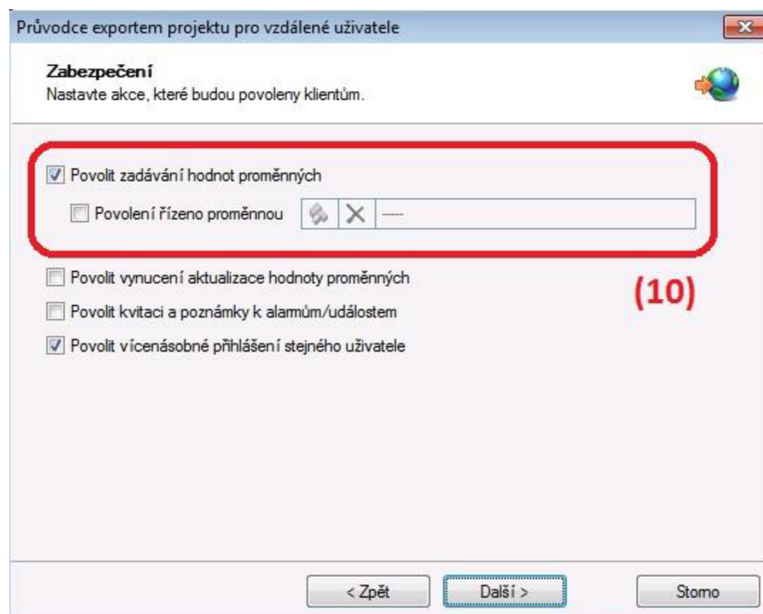
Obr. 4-5 Export pro vzdálené uživatele – 4



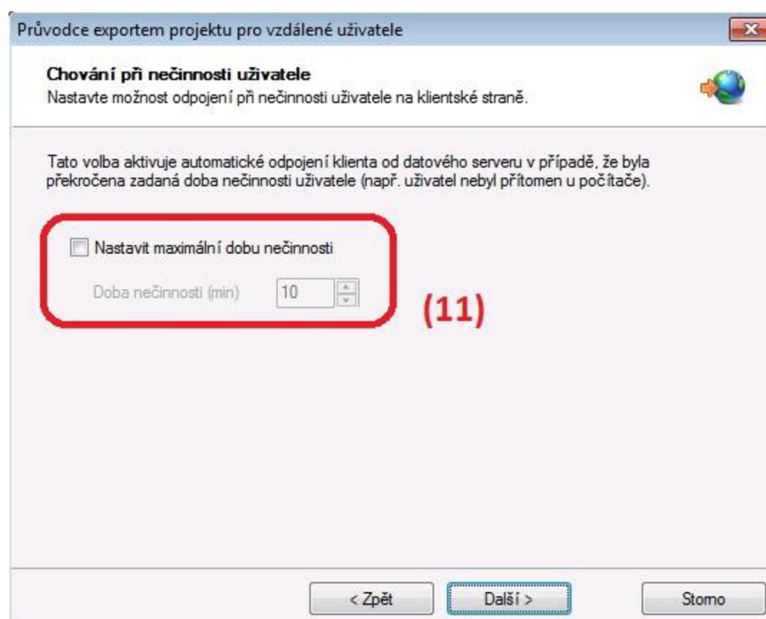
Obr. 4-6 Export pro vzdálené uživatele – 5

Na další stránce (Obr. 4-5) je volba nastavení **rozlišení** projektu pro **PC (6)** a pro **vzdálené uživatele (7)**. Máme zde na výběr z přednastavených rozlišení a možnosti zvolit vlastní rozměr. Na další stránce (Obr. 4-6) můžeme nastavit **Interval aktualizace dat (8)** a maximální interval mezi odesláním a přijetím paketů **Timeout (9)**. Následuje strana (Obr. 4-7), kde si můžeme zvolit, které akce budou povoleny tenkým klientům, např. **povolení zadávání hodnot proměnných, které můžeme řídit proměnnou (10)**. Dle (Obr. 4-8), můžeme nastavit, po jaké době je **klient odpojen při nečinnosti (11)**.





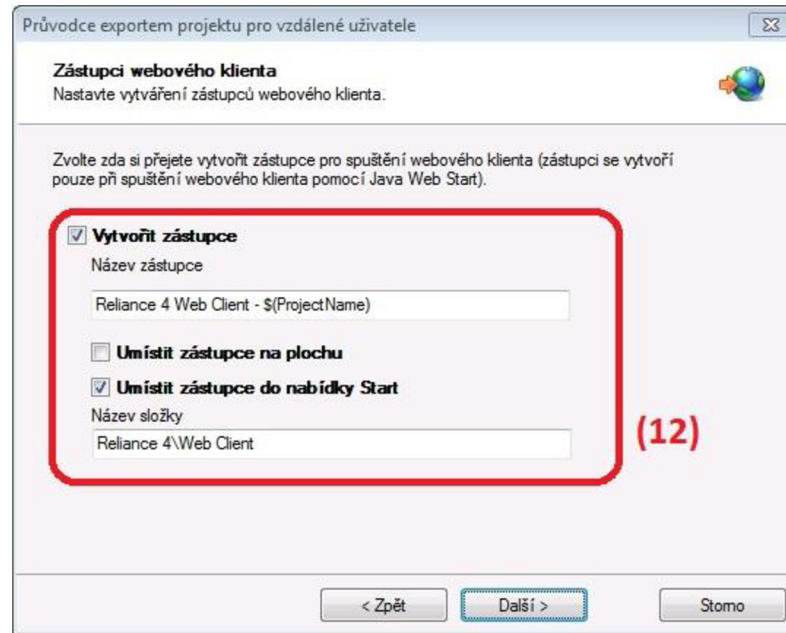
Obr. 4-7 Export pro vzdálené uživatele – 6



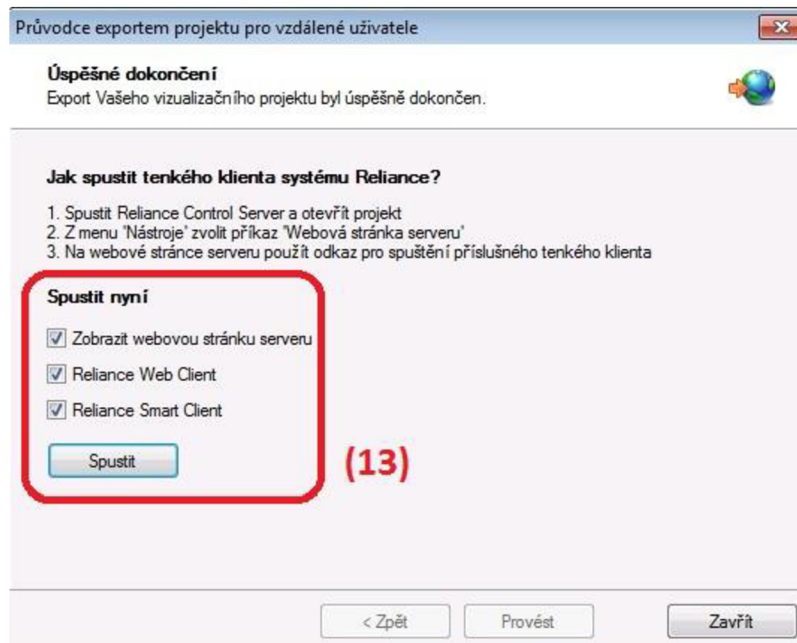
Obr. 4-8 Export pro vzdálené uživatele – 7

Na následující straně (Obr. 4-9), lze nastavit **vytvoření zástupce (12)** pro web klienta. Následuje strana s možností oznámení v případě, že se projekt liší oproti poslední exportované verzi. Na další straně je uvedeno shrnutí projektu a následně je projekt exportován a zobrazena následující strana (Obr. 4-10), je uveden návod na spuštění a příkaz na **spuštění (13)**.

V případě, že potřebujeme exportovat projekt znovu, je možné postupovat dle (Obr. 4-2), **Hlavní menu / Projekt (1) / Znovu exportovat pro vzdálené uživatele (3)**.



Obr. 4-9 Export pro vzdálené uživatele – 8



Obr. 4-10 Export pro vzdálené uživatele – 9

## 4.2 Spuštění vzdálených klientů

Pro spuštění projektu z tenkého klienta je nutné do příkazového řádku prohlížeče na tenkém klientu zadat IP adresu PC, na kterém běží modul Control Server doplněnou o port 40000, např. **192.168.0.150:40000**. V případě nenavázání komunikace, je možné řešení vypnutí firewallu, který brání tenkému klientu se připojit na PC, které je v tu chvíli server.

Pokud je vše dobře nastaveno, naváže se komunikace a v prohlížeči se nám zobrazí webové rozhraní modulu Control Server (*Obr. 4-11*). Tlačítkem **Smart Client (14)** spustíme vizualizační prostředí. Ještě předtím se však zobrazí dotaz na to, zda chceme vizualizaci zobrazit v responzivním nebo normálním zobrazení. U responzivního zobrazení se zobrazení stránky přizpůsobí velikosti displeje zařízení. U normálního zobrazení se zobrazení stránky přizpůsobí rozměrům vizualizace. Výsledná vizualizace vzdáleného řízení z tenkého klienta je uvedena na (*Obr. 4-12*).

**Reliance 4 Control Server**

[Přihlásit uživatele](#) | Nápověda: [CZ](#), [EN](#), [RU](#) | Jazyk: [English](#)

**Vítejte!**  
Vítá Vás webová stránka programu Reliance 4 Control Server.

**Stav**

Název	Hodnota
Verze	4.7.1.25172
Ověření licence	Platnost licence ověřena licenční službou (Port 3323).
Sériové číslo	5B7E-8642
Datové body	500
Projekt	Instalace
Počítač	PC1
Počet připojení (tenci klienti)	3/3
Počet připojení (Reliance OPC Server)	0
Počet připojení (uživatelské aplikace)	0

**Projekt**  
Vyberte součást projektu, kterou chcete zobrazit.

[Stanice a proměnné](#)
[Aktuální alarmy/události](#)
[Tabulkové sestavy](#)
[Uživatelské sestavy](#)

**Tenci klienti**  
Vyberte tenkého klienta, kterého chcete spustit.

[Web Client](#)
[Mobile Client](#)
[Smart Client \(14\)](#)

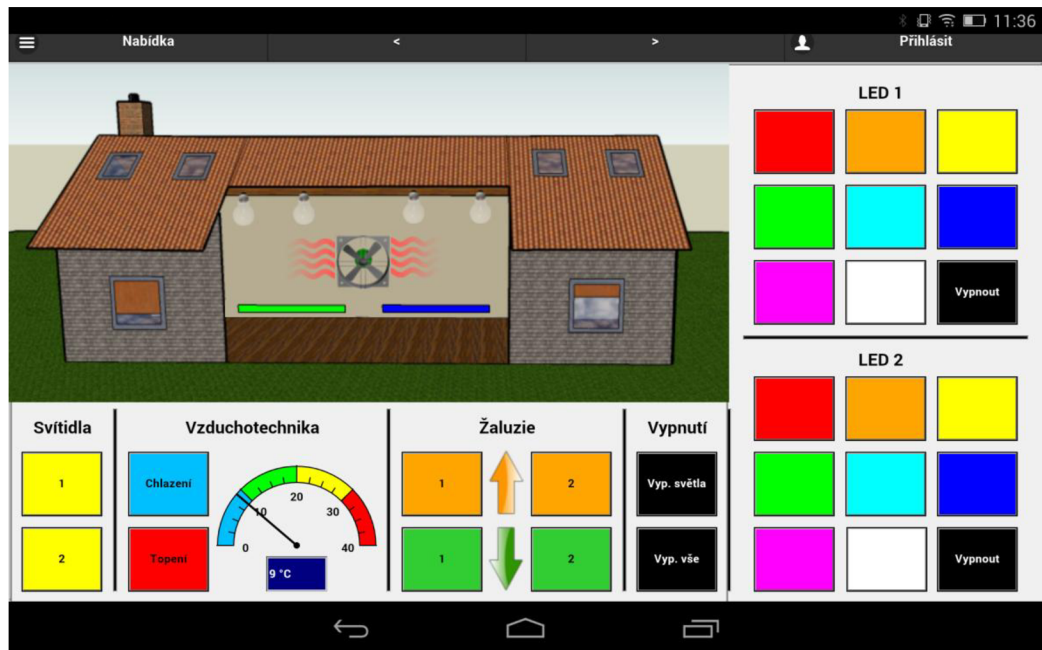
**Server**  
Vyberte položku pro správu serveru.

[Administrace serveru](#)
[Zabezpečené připojení](#)

Copyright © 1997-2015 GEOVAP, spol. s r.o. GEOVAP

Reliance 4 Control Server, 17.5.2016 11:47:50, [www.reliance.cz](http://www.reliance.cz)

*Obr. 4-11 Webové rozhraní*



Obr. 4-12 Vizualizační prostředí tenkého klienta

## 5 ZÁVĚR

Tato práce byla rozčleněna na tři základní oddíly, první oddíl je teoretický a srovnává jednotlivé SCADA/HMI systémy. V druhém oddílu je praktická část, která popisuje tvorbu vizualizačního prostředí pro zadaný panel. V třetí části je vytvořeno vzdálené řízení v reálném čase pro tenkého klienta.

### 5.1 Přehled a srovnání SCADA/HMI systémů

V této části jsou uvedeny tři SCADA/HMI systémy a to konkrétně Reliance 4, Promotic 8.3 a TIRS.NET. Co do struktury systému jsou si systémy Reliance a Promotic podobné. Oba tyto systémy jsou rozčleněny na vývojové a běhové prostředí. U obou systémů jsou tyto prostředí distribuována samostatně. Systém TIRS.NET není členěn na vývojové a běhové prostředí a je tedy distribuován jako celek. Systém Reliance má čtyři běhová a dvě vývojová prostředí oproti systému Promotic, který má pouze jedno běhové a jedno vývojové prostředí.

Všechny tři systémy mají také k dispozici tenkého klienta, přes kterého lze vytvořenou instalaci sledovat či řídit na chytrém telefonu nebo tabletu.

Tyto systémy mají také velkou funkcionalitu s ohledem na komunikační možnosti. U všech třech systémů je dostupná celá řada komunikačních driverů, kterými lze připojit k systému programovatelné automaty, telemetrické stanice, atd. Nejvíce dostupných komunikačních driverů má systém TIRS.NET. V případě nedostupných komunikačních driverů lze využít univerzální komunikační driver, případně OPC server.

Cenové srovnání jednotlivých systémů je poněkud komplikované, jelikož každý systém má jiné licenční členění, např. počet datových bodů, počty klientů, zvláštní ceník pro vývojáře a pro koncové uživatele. Déle jsou nabízeny také zvýhodněné balíčky, které kombinují vývojové a běhové prostředí. Aby bylo možné porovnat výhodnost toho, či druhého systému, bylo by nutné určit příklad modelové aplikace. V (*kap. 2.4*) je uvedeno srovnání cen jednotlivých systémů pro jednotlivé prostředí v závislosti na počtu datových bodů a počtů klientů.

Systém TIRS.NET je oproti druhým dvěma dodáván pro implementační a realizační firmy, výrobce a dodavatele zařízení nebo komponent zdarma.

### 5.2 Praktická aplikace řízení reálné systémové elektroinstalace Foxtrot s vizualizací

V praktické části bylo úkolem vytvořit vizualizační prostředí pro zadaný panel se systémovou elektroinstalací Foxtrot. K tvorbě vizualizačního prostředí byl využit SCADA systém Reliance 4. V této části je popsána podrobná tvorba vizualizačního prostředí, použitých komponent a skriptů. Skripty, které byly v práci využity, slouží k řízení grafických prvků vizualizace nebo k reálnému řízení systémové elektroinstalace.

### 5.3 Vzdálené řízení elektroinstalace v reálném čase

Součástí praktické části bylo vzdálené řízení elektroinstalace v reálném čase. Vzdálené řízení v reálném čase bylo zrealizováno pomocí tenkého klienta. Navržené vizualizační prostředí bylo

---

vyexportováno pro tenkého klienta. Současně byla vytvořena datová komunikace jednotlivých zařízení přes wi-fi router, ke kterému byla přes UTP kabel připojena centrální řídicí jednotka CP-1000. Běhové prostředí systému Reliance 4 bylo realizováno modulem Control Server, který běžel na PC. Jako tenký klient byl využit smart client, který byl zobrazen na tabletu se systémem android. Postup exportu projektu pro tenkého klienta a nastavení datové komunikace je v práci postupně popsán.

## POUŽITÁ LITERATURA

- [1] RELIANCE. Reference. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/success-stories>
- [2] RELIANCE. O nás. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/contact/about-us>
- [3] AUTOMA. Přehled trhu softwaru SCADA. [online]. [cit. 2016-05-17]. Dostupné z: <http://automa.cz/res/pdf/43729.pdf>
- [4] RELIANCE. Vývojové prostředí Reliance 4 Design. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/products/reliance4/development-environment>
- [5] RELIANCE. Runtime moduly systému Reliance 4. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/products/reliance4/runtime-software>
- [6] RELIANCE. Tencí klienti systému Reliance 4. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/products/reliance4/thin-clients>
- [7] RELIANCE. Komunikační drivery. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/products/reliance4/communication-drivers>
- [8] RELIANCE. Reliance OPC Server. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/products/opc/reliance-opc-server>
- [9] RELIANCE. Ceník. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.reliance.cz/cs/pricelist>
- [10] PROMOTIC. MICROSYS, spol. s r.o. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/firm/microsys.htm>
- [11] PROMOTIC. PROMOTIC - Reference aplikací. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/firm/reference/pmreferences.htm>
- [12] PROMOTIC. Co je PROMOTIC. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm>
- [13] PROMOTIC. Runtime licence PROMOTIC. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/PriceList/LicencePmRuntime.htm>
- [14] PROMOTIC. PmData(Web)Client. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/PriceList/LicencePmClients.htm#WebClient>
- [15] PROMOTIC. Komunikace pomocí ovladačů systému PROMOTIC. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/Subsystems/Comm/PmDrivers/Group.htm>
- [16] PROMOTIC. Komunikace přes standardní rozhraní OPC. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPC.htm>
- [17] PROMOTIC. Ceník systému PROMOTIC. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/PriceList/PriceList.htm>
- [18] TIRS. Profil firmy CORAL s.r.o. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.tirs.cz/cs/o-firme>
- [19] TIRS. SCADA/HMI systém TIRS. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.tirs.cz/scadahmi-systemy-tirs/>
- [20] TIRS. Architektura systému TIRS.NET. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.tirs.cz/scadahmi-systemy-tirs/tirs-net-architektura-systemu/>

- [21] HW.CZ. TIRS.NET - vizualizační a řídicí systém založený na technologii Microsoft.NET. [online]. [cit. 2016-05-17]. Dostupné z: <http://www.hw.cz/navrh-obvodu/software/tirsnet-vizualizacni-a-ridici-system-zalozeny-na-technologie-microsoftnet.html>
- [22] TIRS.NET 6 – profesionální SCADA/HMI systém [online]. [cit. 2016-05-17]. Dostupné z: <http://www.tirs.cz/scadahmi-systemy-tirs/tirs-net-6-profesionalni-scadahmi-system/>
- [23] TIRS.NET – seznam komunikovaných zařízení [online]. [cit. 2016-05-17]. Dostupné z: <http://www.tirs.cz/scadahmi-systemy-tirs/tirs-net-seznam-komunikovanych-zarizeni/>
- [24] TIRS. Ceník vizualizačního a řídicího SCADA/HMI systému TIRS.NET. [online]. [cit. 2016-05-17]. Dostupné z: [http://www.coral.cz/wp-content/uploads/2015/09/OL\\_Cenik\\_TIRS.NET\\_6\\_2015\\_v1.1.pdf](http://www.coral.cz/wp-content/uploads/2015/09/OL_Cenik_TIRS.NET_6_2015_v1.1.pdf)
- [25] HUBÁLEK, M. Využití řídicího systému Foxtrot jako Building Management System. Diplomová práce. Brno: Ústav elektroenergetiky FEKT VUT v Brně, 2013, 59 stran.
- [26] TECO. Mosaic - pro vývoj PLC programu dle standardu IEC 61131-3. [online]. [cit. 2016-05-17]. Dostupné z: [http://www.tecomat.com/kategorie-311-mosaic-sw\\_.html](http://www.tecomat.com/kategorie-311-mosaic-sw_.html)
- [27] RELIANCE. Tip pro VBScript: Převod textu alarmu/události na řeč. [online]. [cit. 2016-05-17]. Dostupné z: <https://www.reliance.cz/cs/support/articles/technical/vbscript-tip-converting-alarm-event-text-to-speech>
- [28] TECO. CIB – Modul řízení LED pásků. [online]. [cit. 2016-05-17]. Dostupné z: [http://www.tecomat.com/wpimages/other/DOCS/cze/PRINTS/Cat\\_Foxtrot-CZ-datasheets/Foxtrot-CZ-C-DM-0006M-ULED.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/PRINTS/Cat_Foxtrot-CZ-datasheets/Foxtrot-CZ-C-DM-0006M-ULED.pdf)



## **SEZNAM PŘÍLOH NA CD**

- 1) Diplomová práce v PDF
- 2) Složka s programem v SW Mosaic
- 3) Složka s vizualizačním programem v SW Reliance 4