



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Sledování a záznam stavu osvětlení budovy

## Bakalářská práce

*Studijní program:* B2612 – Elektrotechnika a informatika

*Studijní obor:* 1802R022 – Informatika a logistika

*Autor práce:* **Jan Horáček**

*Vedoucí práce:* Ing. Martin Černík, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Monitoring and recording of state of the lighting in a building

## Bachelor thesis

*Study programme:* B2612 – Electrical Engineering and Informatics

*Study branch:* 1802R022 – Informatics and Logistics

*Author:* **Jan Horáček**

*Supervisor:* Ing. Martin Černík, Ph.D.



**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan Horáček**  
Osobní číslo: **M12000285**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Informatika a logistika**  
Název tématu: **Sledování a záznam stavu osvětlení budovy**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s konkrétním osvětlovacím systémem a popište ho.
2. Naprogramujte systém TIRS.NET tak, aby v uvedeném osvětlovacím systému umožnil sledovat následující stavy svítidel: svítí, nesvítí, aktuální příkon svítidla, porucha svítidla.
3. Pro jednotlivé stavy vytvořte dlouhodobé trendy pro zvolené časové období a zpřístupněte je formou tabulky a grafu a možností dalšího zpracování (www, tisk apod.).

Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30–40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] HABEL, Jiří.: Světlo a osvětlování. Praha FCC Public, 2013, 622 s. ISBN 978 808 653 421-3.
- [2] www.coral.cz. [Online]  
<http://www.coral.cz/cs/scadahmi-sw/tirsnet/informace-o-systemu-tirsnet>.
- [3] www.helvar.com. [Online]  
[http://www.helvar.com/sites/default/files/product\\_datasheets/905\\_Single\\_DALI\\_Subnet\\_Router\\_datasheet\\_iss03.pdf](http://www.helvar.com/sites/default/files/product_datasheets/905_Single_DALI_Subnet_Router_datasheet_iss03.pdf).

Vedoucí bakalářské práce: Ing. Martin Černík, Ph.D.

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: Ing. Jan Masařík

EFektivní OSvětlování s.r.o.

Datum zadání bakalářské práce: 10. října 2015

Termín odevzdání bakalářské práce: 16. května 2016



prof. Ing. Václav Kopecký, CSc.  
děkan



doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2015

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

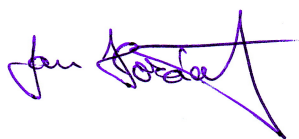
Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 9.9.2016

Podpis:



## **Poděkování**

Rád bych zde poděkoval vedoucímu bakalářské práce Ing. Martinu Černíkovi, Ph.D za cenné rady a odborný dohled, který mi věnoval při řešení dané problematiky. Děkuji také firmě EFektivní OSvětlování s.r.o., konkrétně Ing. Janu Masaříkovi a Radku Nesvadbovi za poskytnutí hardwaru a vývojové licence pro software TIRS.NET.

## **Abstrakt**

Tématem této práce je sledování a záznam stavu osvětlení budovy.

V první části bakalářské práce se nachází stručný popis použitého hardwaru, včetně jeho základních technických údajů. Dále je popsán systém TIRS.NET, který se skládá ze systému SCADA/HMI. Je zde popsán komunikační protokol OPC a protokol DALI.

V praktické části práce seznamuje s postupem vytvoření nového projektu a přidání komunikačního bodu konektoru. Dále je popsána tvorba programu, použití modulů až po konečnou vizualizaci projektu s tvorbou grafů a tabulek.

## **Klíčová slova:**

TIRS.NET, TnOPC, IP, XML, SQL, SCADA/HMI, Helvar, DALI

## **Abstract**

The topic of the bachelor paper is monitoring and record of the conditions of the lightning used in a building.

The first part of the paper describes the hardware used for the purposes of the paper, including its basic technical data. Furthermore, the system TIRS. NET containing SCADA/HMI is focused on. Next, communication protocols OPC and DALI are presented.

The practical part presents the procedure of creation of a new project and adding of a new communication item of the connector. Finally, the creation of the programme, usage of the modules and the final visualization of the project with creation of graphs and charts are dealt with.

## **Key words:**

TIRS.NET, TnOPC, IP, XML, SQL, SCADA/HMI, Helvar, DALI

# Obsah

Úvod.....	13
1 Osvětlovací systém.....	14
1.1 Osvětlovací systém Helvar.....	14
1.1.1 Helvar Imagine Router 920.....	14
1.1.2 LED Driver LC1x30-E-DA .....	16
1.2 Schéma zapojení osvětlovacího systému .....	16
1.3 Elektroinstalace .....	17
1.3.1 Klasická elektroinstalace .....	17
1.3.2 Inteligentní elektroinstalace .....	18
1.4 Řídicí systémy v budovách .....	18
1.4.1 Centralizovaný řídicí systém .....	18
1.4.2 Decentralizovaný řídicí systém.....	19
1.4.3 Hybridní řídicí systém .....	19
2 TIRS.NET.....	20
2.1 Architektura systému TIRS.NET .....	20
2.1.1 SCADA.....	20
2.1.2 HMI.....	20
2.2 Struktura SCADA/HMI systémů .....	21
2.2.1 Server SCADA .....	21
2.2.2 Komunikační síť .....	22
2.2.3 RTU .....	22
2.2.4 MTU.....	22
2.2.5 HMI aplikace .....	22
2.3 Protokol DALI .....	24
2.3.1 Komunikace .....	24
2.4 OPC.....	25
2.4.1 OPC Server .....	25
2.4.2 OPC klient.....	25
3 Praktická část.....	26
3.1 Programování projektu.....	26
3.2 TN Project Studio.....	26
3.2.1 Režimy jádra .....	27



3.3	Vytvoření nového projektu .....	27
3.4	Konektor.....	29
3.4.1	Složení konektoru .....	29
3.4.2	Konfigurace konektoru v projektu .....	29
3.4.3	Přidání PLC.....	31
3.4.4	Přidání bloků a signálu .....	33
3.5	Řízení a sledování stavů svítidel.....	35
3.6	Moduly .....	36
3.6.1	HistorySQL.....	37
3.6.2	Calc .....	39
3.6.3	TN Panel .....	41
4	Zpřístupnění dat.....	49
4.1	Zpřístupnění dat pomocí www.....	49
4.2	Zpřístupnění dat pomocí tisku.....	49
	Závěr .....	50
	Seznam použité literatury .....	52
	Přílohy.....	55
	A Obsah přiloženého CD .....	55

## Seznam obrázků

Obrázek 1: Helvar Imagine 920 .....	15
Obrázek 2: LED Driver LC1x30-E-DA .....	16
Obrázek 3: Schéma zapojení.....	17
Obrázek 4: Architektura SCADA/HMI .....	21
Obrázek 5: Historická data, TirsPanel .....	23
Obrázek 6: Komunikační vodič DALI .....	24
Obrázek 7: Komunikace OPC klient – server.....	25
Obrázek 8: TN Project Studio.....	27
Obrázek 9: Jméno projektu .....	28
Obrázek 10: Umístění projektu.....	28
Obrázek 11: Průvodce konektorem.....	30
Obrázek 12: Import konfigurace.....	30
Obrázek 13: Načtení konfigurace konektoru .....	31
Obrázek 14: Přidání PLC .....	31
Obrázek 15: Prázdňá konfigurace PLC .....	32
Obrázek 16: Adresa OPC.....	32
Obrázek 17: Vyplněná konfigurace PLC.....	32
Obrázek 18: Konfigurace bloků.....	33
Obrázek 19: Konfigurace signálu .....	34
Obrázek 20: Modul HistorySQL.....	37
Obrázek 21: Vytvoření databáze.....	38
Obrázek 22: Nové připojení k databázi .....	38
Obrázek 23: Modul Calc.....	39
Obrázek 24: Přístroje v TN Panelu .....	41
Obrázek 25: Vlastnosti tlačítka.....	42
Obrázek 26: Vlastnosti spouštěné aplikace .....	42
Obrázek 27: Spojení na SQL databázi .....	43
Obrázek 28: Konfigurace SQL připojení.....	43
Obrázek 29: Spouštěč aplikace - příkazová řádka .....	44
Obrázek 30: Signály k zobrazení .....	44
Obrázek 31: Konfigurace osy Y .....	45
Obrázek 32: Konfigurace signálů .....	46

Obrázek 33: Graf spotřeby .....	46
Obrázek 34: Tabulka dat .....	48

## **Seznam tabulek**

Tabulka 1: Parametry Helvar 920 .....	15
Tabulka 2: Parametry LED driver .....	16
Tabulka 3: Konfigurace příznaku bloku .....	33
Tabulka 4: Datové typy .....	35

## **Seznam zdrojových kódů**

Zdrojový kód 1: Zdrojový kód pro výpočet celkové spotřeby .....	40
Zdrojový kód 2: Zdrojový kód pro sledování stavu svítidla .....	40

## Seznam zkratek

SCADA	Supervisory Control And Data Acquisition, supervizní řízení a sběr dat
HMI	Human Machine Interface, rozhraní mezi člověkem a strojem
RTU	Remote Terminal Unit, vzdálená jednotka
MTU	Master Terminal Unit, hlavní jednotka
TCP	Transmission Control Protocol, primární přenosový protokol
IP	Internet Protocol, internetový protokol
PLC	Programmable Logic Controller, programovatelný logický automat
WiFi	Wireless Fidelity, bezdrátová síť
PC	Personal Computer, osobní počítač
LED	Light Emitting Diode, dioda emitující světlo
DALI	Digital Adressable Lighting Interface, digitálně adresovatelné světelné rozhraní
SQL	Structured Query Language, strukturovaný dotazovací jazyk
OPC	OLE for Process Control
DLL	Dynamic link library, dynamicky linkovaná knihovna
XML	Extensible Markup Language, značkovací jazyk
CSV	Comma separated values, hodnoty oddělené čárkami
SW	Software, programové vybavení
BMS	Building management systems, systémy řízení budov

## Úvod

V dnešní době, kdy je kladen stále větší důraz na úsporu energie a snižování nákladů, jsou stále více skloňované termíny řízení a regulace. Z tohoto důvodu se častěji setkáváme v budovách s řídicími systémy. Nedílnou součástí budov jsou systémy pro řízení a sledování stavu osvětlení. Hlavními důvody proč zavádíme osvětlovací systémy, jsou centrální monitoring svítidel, komfort ovládání a zvýšení spolehlivosti svítidel.

Výhodou osvětlovacích systémů je centrální monitoring, který umožňuje sledovat stav svítidel, bez přítomnosti pověřené osoby v objektu, případně rozsvěcet nebo stmívat svítidla. Sledování stavů svítidel nám umožňuje odhalit poruchu svítidla, aniž bychom svítidlo fyzicky zkontrolovali, a tak předejít dalším problémům. Dalším důvodem pro řízení osvětlení je ekonomické hledisko. Řízené osvětlení nám poskytuje vyšší efektivitu práce a nižší náklady na energii. Podle průzkumu společnosti Helvar je v Evropské unii spotřebováno až 14 % energie na osvětlení. Řízením osvětlení lze snížit spotřebovanou energii osvětlení až o 60 %.

Mozkem osvětlovacího systému je řídicí jednotka, která je propojena pomocí sběrnice se světelnými předřadníky. Řídicí jednotka sbírá od připojených zařízení data a předává je nadřazenému systému.

# 1 Osvětlovací systém

Osvětlovací systém je celek složený ze svítidel, předřadníků, které vytváří světelné prostředí v prostoru nebo místnosti. Cílem osvětlovacích systému je vytvoření vhodných světelných podmínek pro zrakový výkon a zrakovou pohodu. Dalším cílem je snížení nákladů na osvětlení. [1] [2]

## 1.1 Osvětlovací systém Helvar

V této bakalářské práci je popsán osvětlovací systém od firmy Helvar. S osvětlovacími systémy od společnosti Helvar je možné dosáhnout úspory energie 40 až 60 % na rozdíl od neřízeného osvětlení. Jedná se o modulární systém pro řízení osvětlení budov, který lze snadno rozšiřovat. Základními stavebními bloky popsaného osvětlovacího systému je Imagine router 920, LED driver a sběrnice DALI. Vzhledem k tomu, že tento systém obsahuje komunikační protokol s otevřenou konektivitou OPC, je možné připojit různé systémy třetích stran pro řízení budov (BMS-Building management systems). [3]

### Výhody systému:

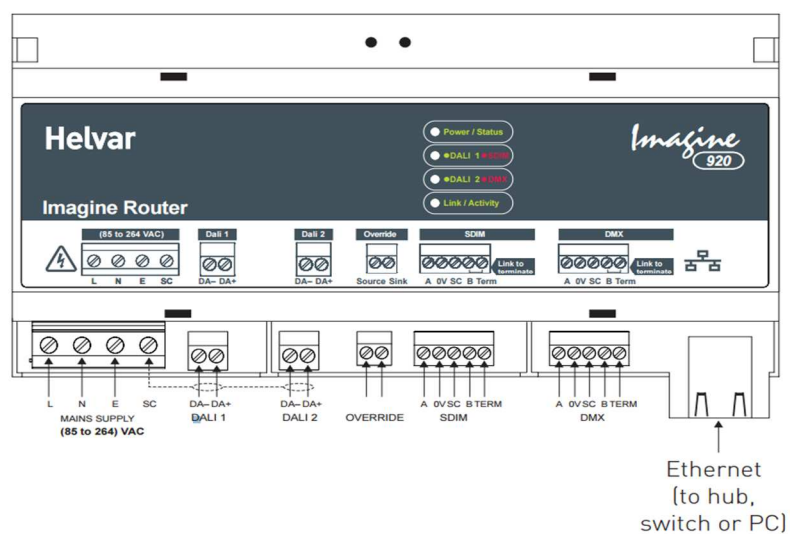
- Je možné vytvořit plně automatizované řízené systémy s pomocí dílů a prvků třetích stran.
- Nejsou potřeba žádné speciální datové kabely.
- Ovládací zařízení komunikují pomocí standardu DALI.
- Prostřednictvím sítě ethernet TCP/IP a protokolu OPC je možné připojit systémy třetích stran. [3]

Pro řízení osvětlovacího systému Helvar bude použit software TIRS.NET, který bude popsán v druhé kapitole.

### 1.1.1 Helvar Imagine Router 920

Router Imagine 920 je PLC od finské firmy Helvar, které slouží k řízení osvětlení. Tento router využívá Ethernet 10/100Mb/s jako páteřní síť pro komunikaci mezi routery a podporuje 128 připojených zařízení DALI, kde pro každou podsíť je zapojeno 64 zařízení. Pro pokročilou konfiguraci a programování je možné použít software Helvar Designer. Router může běžet bez připojeného PC, jelikož data jsou uložena v samotném

zařízení. PC slouží pro konfiguraci nebo diagnostiku zařízení. Systém umožňuje úporu energie pomocí nastavení světelných funkcí, např. detekce světla.



**Obrázek 1:** Helvar Imagine 920 [4]

**Tabulka 1:** Parametry Helvar 920 [4]

<b>Napájení:</b>	85 V – 264 V, 45 Hz – 65 Hz
<b>Spotřeba el. energie:</b>	23 VA
	DALI 2 x 250 mA
<b>Sběrnice DALI:</b>	Vodič 0,5 mm <sup>2</sup> – 2,5 mm <sup>2</sup> , Maximální délka: 300 m při průřezu vodiče 1,5 mm <sup>2</sup>
<b>Přenos dat DALI:</b>	DALI standard IEC 62386
<b>Ethernet:</b>	1 × RJ45 10/100 Mb/s, Cat 5E do 100 m
<b>Izolace:</b>	4 kV
<b>Krytí:</b>	IP30, pro konektory IP00
<b>Pracovní teplota:</b>	0 °C – 40 °C
<b>Rozměry (š × v × h):</b>	160 mm x 58 mm x 100 mm
<b>Hmotnost:</b>	260 g
<b>Montáž:</b>	DIN lišta

### 1.1.2 LED Driver LC1x30-E-DA

Předřadník Helvar LC1x30-E-DA slouží pro řízení a napájení LED svítidel. Tento LED driver patří do kategorie stmívatelných předřadníků a je určen pro vysocejasové LED. Obsahuje svorkovnici pro sběrnici DALI. Driver umožňuje několik způsobů stmívání: logaritmické, lineární nebo stmívání podle vlastního výběru. [5] [6]



Obrázek 2: LED Driver LC1x30-E-DA [7]

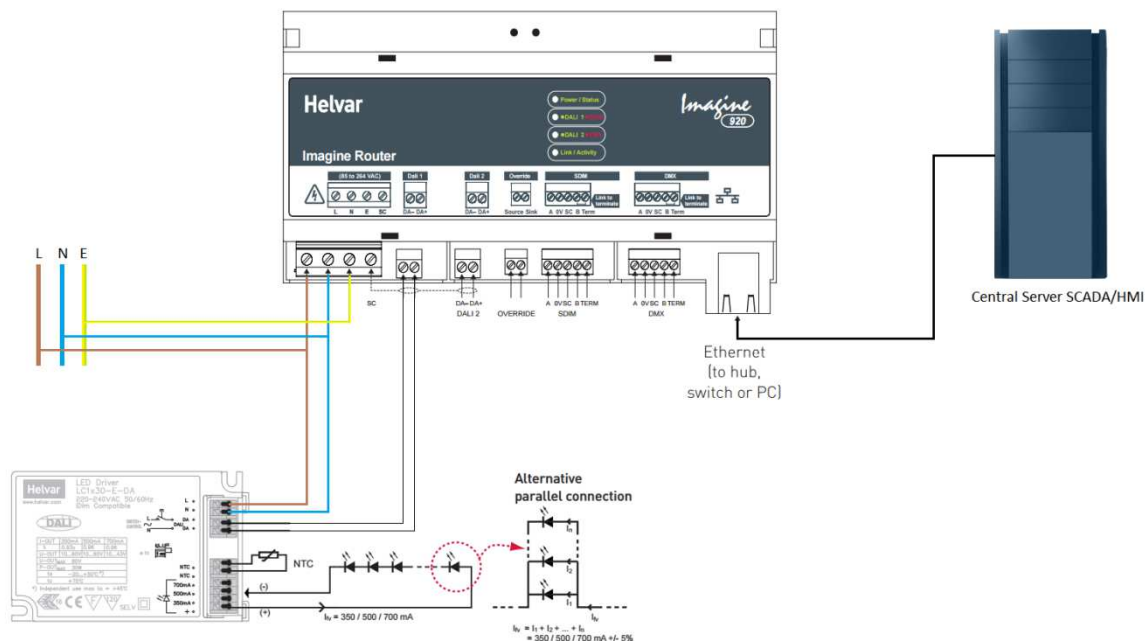
Tabulka 2: Parametry LED driver [6]

<b>Napájení:</b>	198 V – 264 V, 50 Hz – 60 Hz
<b>Max. proud při plném zatížení</b>	0,14 – 0,18 mA
<b>Max.zatížení</b>	30 W
<b>Teplota okolí při provozu</b>	- 20 °C .. + 50 °C
<b>Rozsah stmívání</b>	1 % - 100 %
<b>Rozměry (š × v × h):</b>	106 mm x 67 mm x 28 mm

## 1.2 Schéma zapojení osvětlovacího systému

Na obrázku 7 je zobrazeno schéma zapojení centrálního serveru SCADA/HMI s osvětlovacím systémem Helvar. Centrální server komunikuje s routerem přes rozhraní ethernet, pomocí protokolu OPC. Server sbírá data z routeru a zasílá povely k řízení. K routeru je připojen LED driver, který komunikuje přes sběrnici DALI.





**Obrázek 3:** Schéma zapojení

V případě nasazení aplikace je možné rychlé rozšíření aplikace díky modulární instalaci a rychlému rozšíření aplikace. Přidání LED předřadníku do elektroinstalace je snadné a není nutné přidávat další kabely do elektroinstalace.

### 1.3 Elektroinstalace

V dnešní době se v budovách využívá stále větší množství přístrojů a zařízení. Podle použití těchto přístrojů dělíme elektroinstalace na dvě skupiny:

- klasická elektroinstalace
- systémová elektroinstalace [8]

#### 1.3.1 Klasická elektroinstalace

Klasická elektroinstalace se skládá ze samostatných obvodů, které umožňují zapínat obvod osvětlení, topení nebo obvod, kterým jsou ovládány rolety. Zapojení této instalace je neměnné, silové kabely jsou zasekány do zdi. Úprava této instalace, znamená vyšší náklady a zednické práce. Klasická elektroinstalace neslouží pro přenos dat, ale pro silové spínání spotřebičů. [8]

### **1.3.2 Inteligentní elektroinstalace**

Inteligentní elektroinstalace slouží k řízení technologií v budově. Hlavním úkolem této instalace je sjednotit samostatné technologie do jednoho celku. Celý systém následně řídí a reguluje technologie, jako jsou topné systémy, větrací systémy, ovládání a řízení osvětlení, řízení rolet a následně vizualizaci uživateli. Inteligentní instalace je modulární instalace, ke které se připojují prvky pomocí sběrnice. Instalační sběrnice obsahuje dva vodiče a používá se pro přenos dat a napájení připojených zařízení. V inteligentní instalaci se pomocí ovládacího prvku nespíná příkon do spotřebiče, ale posílají se po sběrnici povely pro spínání. Sběrnice je napájena malým napětím. Díky sběrnici lze systém snadno rozšířit o další prvky.

Hlavním důvodem proč zavádíme inteligentní elektroinstalace je komfort ovládání, řízení a snadné rozšíření systému. Nevýhodou této instalace je finanční náročnost u malých nebo jednoduchých systémů. [8] [9]

## **1.4 Řídicí systémy v budovách**

V budovách se můžeme setkat se třemi druhy řídicích systémů:

- centralizovaný řídicí systém,
- decentralizovaný řídicí systém,
- hybridní řídicí systém.

### **1.4.1 Centralizovaný řídicí systém**

U centralizovaného systému jsou ovládací prvky a spotřebiče propojeny do hvězdy s centrální řídicí jednotkou. Každý spotřebič nebo senzor má vlastní propojení s řídicí jednotkou a vzájemně komunikují přes tuto řídicí jednotku. Tento řídicí systém je obvyklý pro programovatelné automaty PLC. Nevýhodou je hvězdicová topologie, u které při výpadku centrálního prvku přestane fungovat celý systém. [8]

### **1.4.2 Decentralizovaný řídicí systém**

Decentralizovaný systém je takový systém, u kterého má prvek „inteligenci“, prvek tedy disponuje mikroprocesorem a pamětí. Každý prvek je připojený na sběrnici a neexistuje zde žádné centrální řízení. Díky neexistenci centrálního řízení v případě výpadku nějakého prvku není ohrožen chod celého systému, ale jen konkrétní větev, u které se vyskytla porucha. Systém má větší spolehlivost. [10]

### **1.4.3 Hybridní řídicí systém**

Hybridní systém spojuje výhody centralizovaného a decentralizovaného systému. U tohoto systému jsou vstupy zapojeny na sběrnici, ale výstupy jsou připojeny k řídicí jednotce. [11]

## **2 TIRS.NET**

System TIRS.NET je systém z kategorie SCADA/HMI, který slouží k řízení a monitorování technologií. Je určen převážně pro nasazení jednodušších aplikací, kde je kladen důraz na rychlost a jednoduché řešení. [12]

### **2.1 Architektura systému TIRS.NET**

TIRS.NET se skládá obecně ze dvou základních částí. Serverová část (SCADA), je pro uživatele neviditelná a běží jako služba systému Windows. Klientská část (HMI) je viditelná pro uživatele, jako jsou vizualizační panely, konfigurační moduly nebo připojené technologie. [12]

#### **2.1.1 SCADA**

SCADA je zkratka pro Supervisory Control And Data Acquisition, v překladu to znamená „supervizní řízení a sběr dat“, jedná se o software, který se zaměřuje na úroveň supervizora, není tedy plnohodnotným systémem, ale zpravidla funguje nad skutečným řídicím systémem jako je PLC. SCADA tvoří serverovou část systému, kde běží jádro systému TIRS.NET - TirsNetKernel.exe, komunikační konektory a moduly. [13] [14]

#### **2.1.2 HMI**

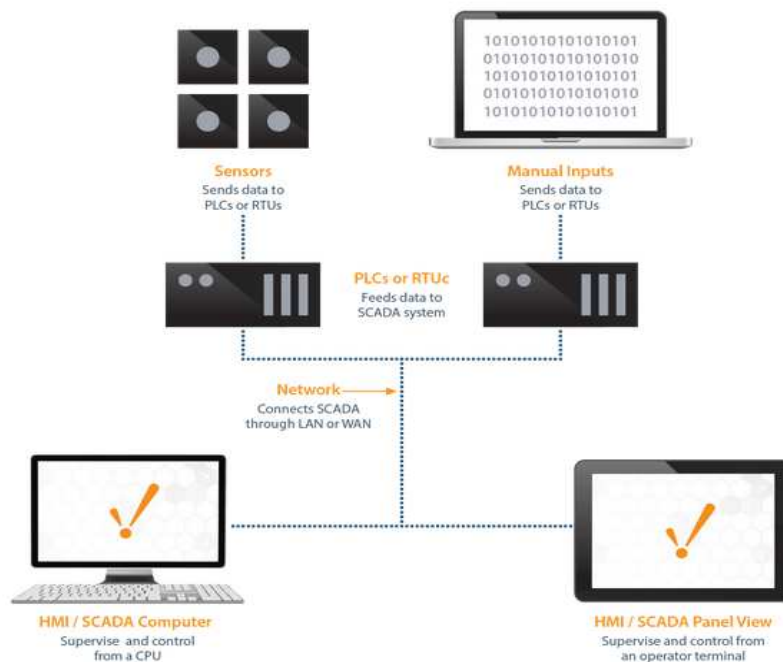
HMI znamená Human Machine Interface, tj. „rozhraní mezi člověkem a strojem“. HMI je klientská část systému, která umožňuje zobrazovat panely, monitorovat a řídit připojené technologie pomocí vizualizačních panelů. Pomocí klientské aplikace je možné vytvářet a editovat vizualizační panely. Operátorovi umožňuje zobrazovat informace o vybraných veličinách, zadávat operátorské povely a zobrazovat data z databází. [14]

## 2.2 Struktura SCADA/HMI systémů

Systém SCADA/HMI se skládá z 5 základních komponent:

- server SCADA,
- komunikační síť,
- jednotky RTU,
- MTU,
- HMI.

Struktura systému SCADA/HMI je zobrazena na obrázku 1.



**Obrázek 4:** Architektura SCADA/HMI [15]

### 2.2.1 Server SCADA

Server je označení pro počítač, který poskytuje určité služby nebo je to software, který realizuje služby. Centrální server umožňuje provoz SCADA softwaru, který se stará o monitorování, řízení a sběr dat. SCADA poskytuje rozhraní pro HMI, které umožňuje vizualizaci funkcí, změnu různých parametrů a zobrazení dat, hodnot signálů či zobrazení alarmových hlášení. [16] [17]

### **2.2.2 Komunikační síť**

Komunikační síť je spojnice dvou nebo více bodů, přes které komunikují zařízení. Důležitým požadavkem na komunikační síť je zajištění nepřetržité a obousměrné komunikace mezi jednotkami RTU a MTU. Vysoká spolehlivost je požadovaná na sběr dat. Pro komunikační síť můžeme využít metalické a optické spoje nebo WiFi technologii, která není tak stabilní z pohledu ztrátovosti paketů. Pro použití těchto technologií je nutné použít komunikační protokol TCP/IP. [16] [17]

### **2.2.3 RTU**

RTU je zkratka pro Remote Terminal Unit. Tato jednotka shromažďuje nasbíraná data do paměti z připojených zařízení, dokud MTU nezašle příkaz k odeslání dat. Propojuje tedy jednotlivá fyzická zařízení, jako jsou ventily, čerpadla a čidla s jednotkou MTU, která je na druhé straně. Ve většině případů jsou RTU realizovány jako programovatelné automaty PLC, které jsou schopny řešit jednodušší úkoly samostatně bez jednotky MTU. PLC obsahuje procesor, který je schopen přijímat data z komunikačních zařízení, které následně zpracuje. Komunikace mezi RTU a MTU probíhá pomocí sběrnice nebo datové sítě. Komunikační protokol je uzavřený nebo otevřený. Otevřené komunikační protokoly jsou Modbus a TCP/IP. [16] [17]

### **2.2.4 MTU**

MTU je zkratka pro Main Terminal Unit. Tato jednotka bývá součástí SCADA serveru, která zajišťuje veškerou komunikaci s jednotkou RTU a rozhraním pro operátora. Data ze vzdálených zařízení jsou posílána prostřednictvím RTU do jednotky MTU, kde se zpracují a následně se zasílají dalším systémům. [16] [17]

### **2.2.5 HMI aplikace**

HMI aplikace poskytují operátorovi dva základní modely:

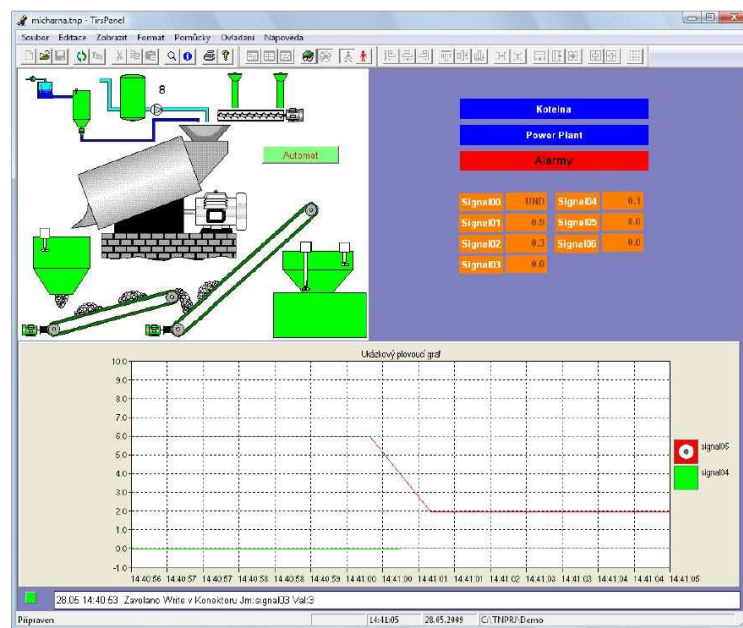
- vizualizace technologie,
- historická data.

### 2.2.5.1 Vizualizace technologie

Vizualizace operátorovi zobrazuje stav technologie pomocí grafických schémat a přehledů. Ve vizualizačním panelu je možné monitorovat nebo řídit danou technologii. Možností vizuálních technologií je mnoho, můžeme např. spouštět různé moduly a nástroje. Tato část je pro operátora důležitá, jelikož mu může zobrazit stav technologií, alarmy, nekomunikující technologie nebo např. zhasnuté či rozsvícené osvětlení místností.

### 2.2.5.2 Historická data

Zobrazování historických dat je důležité z hlediska analýzy provozu zařízení, sledování odběru elektrické energie nebo sledování procentuální úrovně osvětlení různých místností. Mezi historická data řadíme zobrazení tabulek a grafů.



Obrázek 5: Historická data, TirsPanel

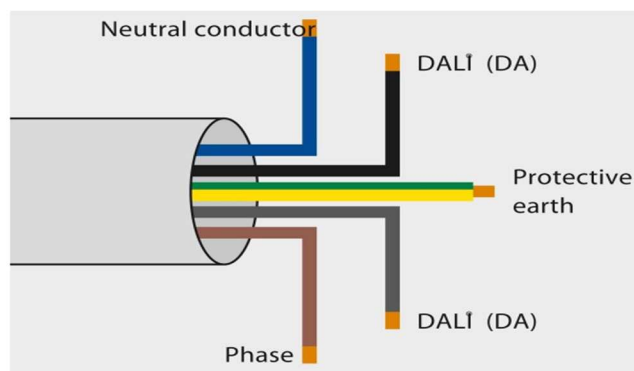
## 2.3 Protokol DALI

DALI je digitálně adresovatelné světelné zařízení, které je standardem pro elektronické předřadníky, pro hromadné ovládání svítidel, umožňuje snadné a inteligentní řízení osvětlení. DALI je otevřený protokol, který se definuje podle normy IEC 62386. Na sběrnici můžeme připojit až 64 předřadníků, které mohou být přiřazeny až do 16 skupin. [18] [19]

DALI, umožňuje plynulé stmívání a monitoring stavů svítidel. Protokol využívá obousměrnou komunikaci mezi řídicí jednotkou a svítidlem. Komunikace mezi svítidlem a řídicí jednotkou může probíhat po fázovém vodiči nebo bezdrátově. U bezdrátové varianty je nutné instalovat do světla přijímač pro příjem signálu. U komunikace po fázovém vodiči musí být instalován převodník signálu. [20]

### 2.3.1 Komunikace

System DALI, je typu Master-Slave. V tomto systému posílají řídicí moduly příkazy předřadníkům, které jsou připojené ke svítidlům. Master umožňuje komunikaci s dalšími zařízeními typu Master. Slave odpovídá nebo vykonává instrukce jen na příchozí dotazy. Ke komunikaci využívá sběrnice dvou vodičů nebo speciální kabely, které obsahují DALI1, DALI2, L, N a PE vodiče. Definované napětí pro log 0 je v rozmezí -6,5 V, až 6,5 V. Pro log 1 je definované napětí v rozsahu 9,5 V až 22,5 V, ale standardní napětí pro log 1 je obvykle 16 V. [19] [21]



**Obrázek 6:** Komunikační vodič DALI [22]



## 2.4 OPC

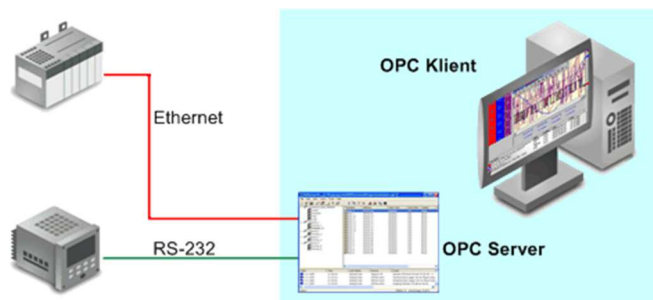
OPC neboli OLE Process Control, je komunikační protokol pro propojení SW aplikací a hardwaru pro řízení technologií. Tento protokol je založen na architektuře klient-server. Standard OPC nám zajišťuje kompatibilitu hardwaru a softwaru. Komunikační kanálem je RS-232 nebo Ethernet. [23]

### 2.4.1 OPC Server

OPC server je softwarový program, který komunikuje s připojenou technologií určitým komunikačním protokolem (Modbus, TnOPC atd.). Získaná data se převádí do standardního formátu OPC a následně jsou poskytnuta OPC klientu k dalšímu zpracování. [23]

### 2.4.2 OPC klient

OPC klient je softwarový program, který přijímá data z OPC serveru a poskytuje tato data uživateli k zobrazení grafů, tabulek a dalších reportů.



**Obrázek 7:** Komunikace OPC klient – server [23]

### **3 Praktická část**

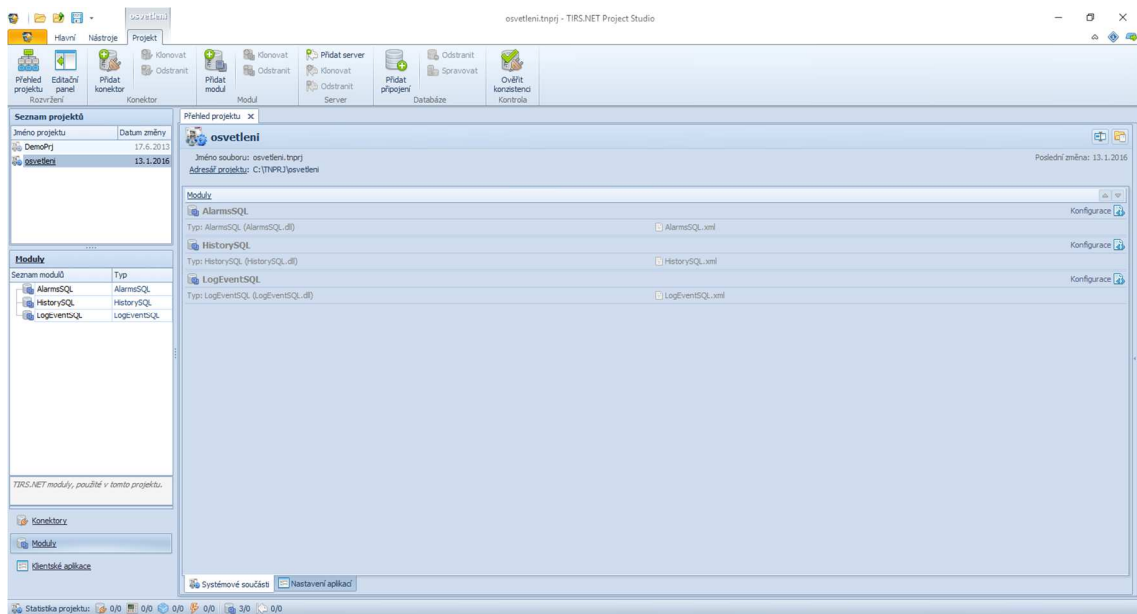
V této bakalářské práci jsem se bohužel nedostal ke konkrétní budově, ve které by byl instalovaný osvětlovací systém. Osvětlovací systém lze snadno a rychle rozšířit díky modulárnímu osvětlovacímu systému Helvar a řídicímu systému TIRS.NET.

#### **3.1 Programování projektu**

Jedním z cílů bakalářské práce je naprogramovat systém TIRS.NET. V této části bakalářské práce bych chtěl nastavit a zprovoznit systém, kterým by bylo možné ovládat osvětlení. Dále, aby umožňoval sledování stavů svítidel (svítí / nesvítí a porucha svítidla), tak aby tyto hodnoty bylo možné ukládat do databáze k dalšímu zpracování. Další částí, kterou bych chtěl vyřešit je aktuální a celkový příkon svítidel, který je daný součtem svítidel. V poslední části této práce se zaměříme na uložená data v databázi. Data z databáze se budou zobrazovat do grafů a tabulek, které se následně publikují na web.

#### **3.2 TN Project Studio**

Nástroj TN Project Studio je základní součástí systému TIRS.NET. Tento nástroj je určen pro navrhování nových projektů nebo pro editaci stávajících projektů. V TN Project studiu můžeme definovat komunikační konektory pro komunikaci s příslušným hardwarem. V našem případě použijeme komunikační konektor TnOPC. Dále je možné konfigurovat moduly (HistorySQL, Calc atd.). Studio umožňuje přepínat chod jádra mezi režimy ladění a služby.



Obrázek 8: TN Project Studio

### 3.2.1 Režimy jádra

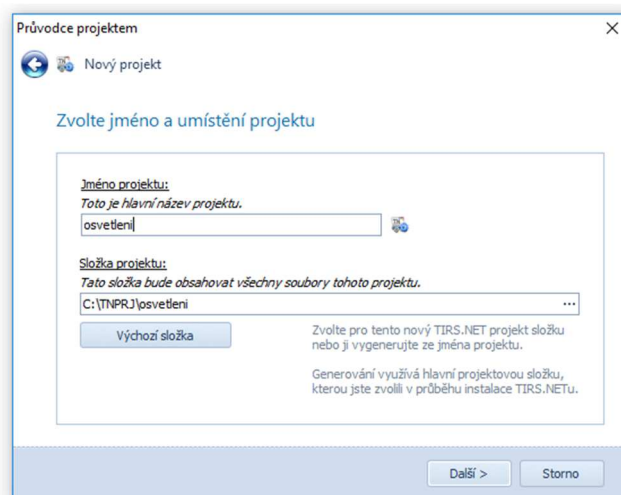
Jádro systému může pracovat v režimu služby nebo v režimu ladění.

**Režim ladění** – tento režim je určen pro ladění projektu. Jádro systému se spouští, pokud se spustí libovolná klientská aplikace systému TIRS.NET. [31]

**Režim služby** – tento režim se používá pro dispečink, kde již je aplikace odladěná. Režim se spouští automaticky jako služba systému Windows. [31]

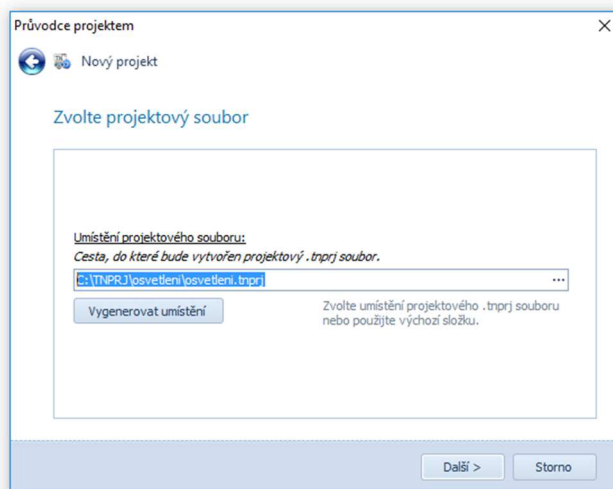
### 3.3 Vytvoření nového projektu

Otevřeme si software TN Project Studio a v horní liště vybereme z možností Hlavní → Nový projekt. Zobrazí se nám průvodce projektu TIRS.NET. Pro pokračování stiskneme tlačítko další. V následující části je nutné zadat hlavní název projektu a výchozí složku projektu.



**Obrázek 9:** Jméno projektu

Pro pokračování stiskneme tlačítko další a zobrazí se nám okno s cestou, do které bude vytvořen nový projektový soubor osvetleni.tnprj. V další části je již rekapitulace projektu, kde vidíme jméno projektu, projektový soubor a složku umístění projektu.



**Obrázek 10:** Umístění projektu

V poslední fázi se zobrazí dialogové okno s textem „Přidat tento projekt do Seznamu projektů?“, volbu potvrdíme možností ano, jinak by nebyl projekt vidět v TN Project Studiu. Tímto jsme dokončili založení prázdného projektu v programu.

### 3.4 Konektor

Konektor je SW modul, který slouží pro komunikaci systému TIRS.NET a připojené technologie. Pro komunikaci s připojenou technologií je nutné mít v počítači umístěnou komunikační knihovnu, neboli konektor, který odpovídá pro dané PLC. [24]

Po přidání konektoru TnOPC, který slouží pro komunikaci mezi hardwarem a softwarem nastává problém v komunikaci. Tento konektor nekomunikuje úplně korektně s připojeným hardwarem. Problém způsobuje výpadky komunikace v náhodném časovém okamžiku. Výpadek komunikace způsobuje odpojení routeru, který je následně nutné znovu připojit. V současné chvíli neexistuje jiný komunikační konektor pro systém TIRS.NET, který by nahradil stávající TnOPC konektor a byl schopen komunikovat se zařízením od společnosti Helvar.

#### 3.4.1 Složení konektoru

Konektor se skládá ze tří částí – společné funkce jádra, vlastní konektor a konfigurace signálu.

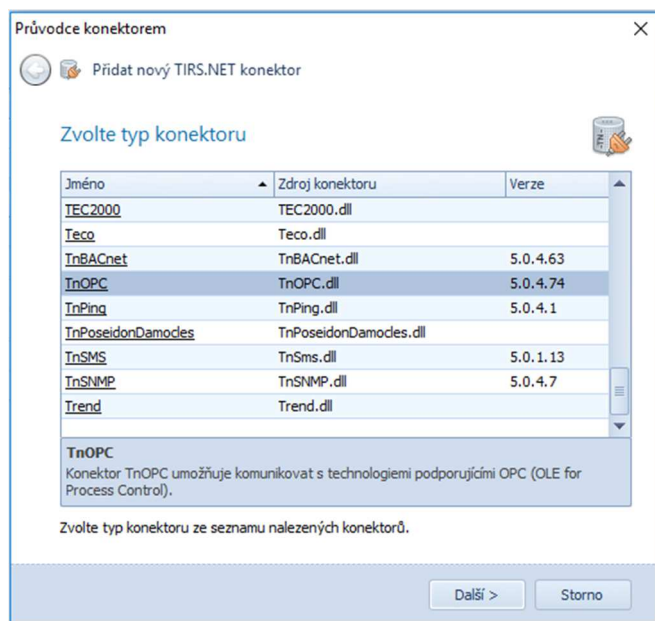
**Společné funkce jádra** zajišťuje základní funkce pro konektory. Jádro má na starosti datovou komunikaci konektoru, správu signálů, čtení dat, zpracování a odesílání dat. [24]

**Vlastní konektor** obsahuje DLL knihovnu, která zahrnuje naprogramovaná komunikační protokol pro danou technologii. [24]

**Konfigurace signálu** je definice proměnných. Signály a parametry mohou být uloženy v souboru XML nebo v databázi. Pro načtení těchto parametrů do jádra systému se používají DLL knihovny. [24]

#### 3.4.2 Konfigurace konektoru v projektu

Pro přidání konektoru do projektu je nutné v navigačním menu zvolit tlačítko „Přidat konektor“, poté se nám zobrazí nabídka „Průvodce konektorem“.



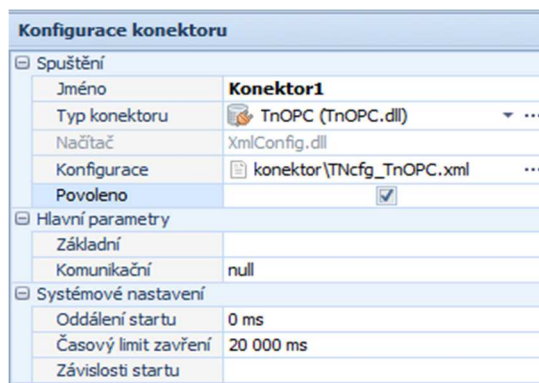
**Obrázek 11:** Průvodce konektorem

V této nabídce vybereme typ konektoru TnOPC a stiskneme tlačítko další. V následující nabídce pojmenujeme konektor a stiskneme dokončit. Takto jsme do projektu přidali nový konektor TnOPC. Po vytvoření nového konektoru je nutné přiřadit konektoru konfiguraci. Pro vytvoření konfigurace konektoru otevřeme soubor TNcfg\_TnOPC\_Empty.ods pomocí aplikace Open Office, který se nachází v adresáři C:\\TNPRJ\_Connectors\_Config.

Export / Import	Exportovat (ano/ne)	Adresa PLC	Flagy (příznaky) PLC	Popis PLC	Min. perioda hodnot [ms]	Navazování spojení	Max. doba připojení OPC serveru	Čas čekání před novým připojením OPC-serveru [ms]	Adresa OPC serveru	Jméno OPC serveru nebo jeho CLSID	Čas čekání po novém připojení OPC-serveru [ms]
Jméno PLC	export	address	flags	prefix	lpar1	lpar2	lpar3	lpar4	spar1	spar2	spar3
name											

**Obrázek 12:** Import konfigurace

Pomocí tlačítka Export/Import vyexportujeme XML konfiguraci, kterou aplikujeme jako prázdnou konfiguraci konektoru.

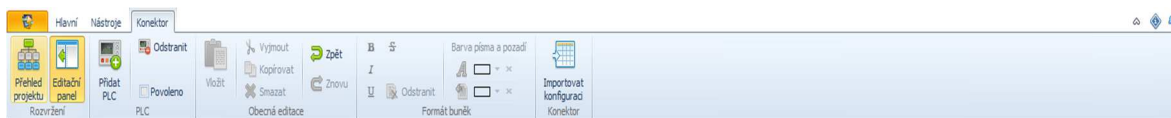


**Obrázek 13:** Načtení konfigurace konektoru

Soubor XML vložíme do projektu tak, že na řádku konfigurace nastavíme cestu k souboru Tncfg\_TnOPC.xml, který jsme si vyexportovali.

### 3.4.3 Přidání PLC

Pro komunikaci s připojeným hardwarem je nutné nastavit komunikaci v TN Project Studiu. Pro přidání PLC si otevřeme aplikaci TN Project Studio. Nejprve si musíme vybrat ze založených projektů a poté můžeme přidat konfiguraci pro připojený hardware. V navigačním menu vybereme volbu „konektor“ a následně vybereme tlačítko „Přidat PLC“.



**Obrázek 14:** Přidání PLC

Nyní se nám v konektor1 zobrazila konfigurace pro PLC zařízení. Pro správné nakonfigurování této konfigurace je nutné nastavit:

**Jméno** – označuje jednoznačný identifikátor v dané konfigurační tabulce.

**Adresa PLC** – určuje jednoznačný identifikátor PLC v komunikačním konektoru.

**Min. perioda hodnot** – vymezuje nejmenší periodu [ms], ve které mohou být poslány změny hodnot.

**Navazování spojení** - stanovuje interval, jak často má TIRS.NET komunikovat se zařízením. Pro stále spojení zvolíme možnost 0 nebo pro spojení jen jednou vybereme variantu 1.

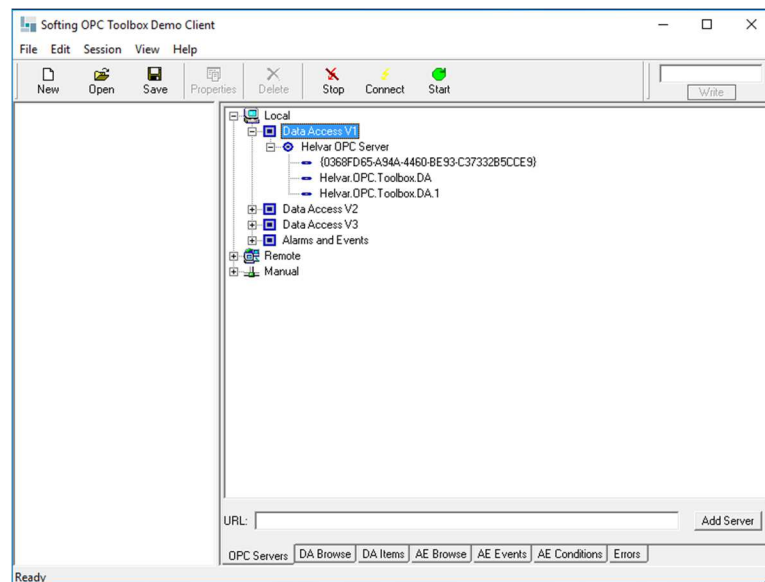
**Adresa OPC serveru** – definuje jméno nebo IP adresu počítače, na kterém běží OPC server.

**Jméno, nebo CLSID OPC** – určuje jméno nebo CLSID OPC serveru.

Seznam PLC											
	Jméno (NAME)	Adres... (ADDR...)	Prefix (PREFIX)	Příznaky (FLAGS)	Min. perioda hod... (LPAR1)	Navazování spoj... (LPAR2)	Max. doba přípoj... (LPAR3)	Čekání před příp... (LPAR4)	Adresa OPC ser... (SPAR1)	Jméno, nebo CL... (SPAR2)	Čekání po přípoj... (SPAR3)
▶	Plc #001	1									

**Obrázek 15:** Prázdná konfigurace PLC

Jméno OPC serveru můžeme zjistit aplikací OPC Demo Client. Po rozbalení položky **Data Access V1** a následné položky **Helvar OPC Server** se nám zobrazí adresa OPC serveru, která je v tomto případě {0368FD65-A94A-4460-BE93-C37332B5CCE9}.



**Obrázek 16:** Adresa OPC

V konečné konfiguraci PLC zařízení máme vyplněno jméno zařízení, příznaky, navazování spojení, adresu OPC serveru a jméno OPC serveru.

Seznam PLC											
	Jméno (NAME)	Adres... (ADDR...)	Prefix (PREFIX)	Příznaky (FLAGS)	Min. perioda hod... (LPAR1)	Navazování spoj... (LPAR2)	Max. doba přípoj... (LPAR3)	Čekání před příp... (LPAR4)	Adresa OPC ser... (SPAR1)	Jméno, nebo CLSID OPC (SPAR2)	Čekání po přípoj... (SPAR3)
▶	PLC_osvetleni	1	PLC1	4	1000	0			localhost	{0368FD65-A94A-4460-BE93-C37332B5CCE9}	

**Obrázek 17:** Vyplněná konfigurace PLC



### 3.4.4 Přidání bloků a signálu

Pro přidání signálů je nutné nejdříve přidat blok. Blok není fyzickým zařízením v PLC, ale jednotkou, která obsahuje signály, které jsou načítány ve stejný čas. [24]

Pro přidání bloku spustíme TN Project Studio a v menu zvolíme záložku konektor a následně tlačítko „Přidat blok“. U nového bloku je nutné vyplnit jméno, příznaky a periodu komunikace.

**Tabulka 3:** Konfigurace příznaku bloku [24]

Název příznaku	číslo bitu	hodnota
Uchovat registraci	bit č. 7	128
Blokové čtení	bit č. 8	256
Perzistentní	bit č. 9	512
Nevyhlašovat FAIL	bit č. 10	1 024
Systémový blok	bit č. 11	2 048
Null při chybě	bit č. 12	4 096

Na obrázku 18 je zobrazena konfigurace bloků v TN Project Studiu. Příznak, který je roven 0, nemá nastaven žádný příznak.

	Jméno (NAME)	Číslo ... (NUM...)	Příznaky (FLAGS)	Perioda komunikace (REFRESHMS)	Zotav... (RECO...)	Čekán... (TIME...)	Počet ... (REPEA...)	Pevný čas (TIMEMASK)	Oddálení (LATER)	Blok: Zdroj dat (LPAR1)	Opakování - čas ... (LPAR2)	Opakování - ma... (LPAR3)	Opakování - ček... (LPAR4)
	Signal	1	0	1000									
	Missing	2	0	1000									
	Failure	3	0	1000									
I	Calc	4	0	1000									

**Obrázek 18:** Konfigurace bloků

### Přidání signálu

Signály předávají modulům a klientům informace o připojených technologiích. Signály rozdělujeme na systémové a externí. Systémové signály jsou speciální signály systému TIRS.NET bez komunikace. Externí signály získávají hodnotu z naměřených veličin z připojených technologií.

Pro přidání signálu do projektu zvolíme konektory a následně vybereme blok, do kterého chceme signál přidat. Poté v menu zvolíme položku konektor a položku „Přidat signál“. Takto jsme do projektu přidali neaktivní signál.

## Konfigurace signálu

Na obrázku 19 je vidět konkrétní konfigurace signálu světlo. Do této konfigurace je nutné vyplnit jméno, příznaky signálu, datový typ, adresu datového bodu, moduly a příznaky pro historii.

Obrázek 19: Konfigurace signálu

**Jméno** - každý signál musí mít jedinečné jméno v rámci celého projektu.

**Popis signálu** – uživatelský popis signálu, který se zobrazuje v přehledu alarmů.

**Povoleno** – každý signál je pro použití potřeba nejprve povolit.

**Datový typ** – určuje datový typ v PLC.

**Tabulka 4:** Datové typy

Označení	hodnota	cilový typ VARIANT
CHAR	10	VT_I1
BYTE	11	VT_UI1
SHORT	12	VT_I2
WORD	13	VT_UI2
LONG	14	VT_I4
DWORD	15	VT_UI4
INT64	16	VT_I8
WORD64	17	VT_UI8
FLOAT	18	VT_R4
STRING	40	VT_BSTR

### 3.5 Řízení a sledování stavů svítidel

Inteligentní elektroinstalace osvětlení je propojena pomocí sběrnice systému. Tato instalace nám snadno umožňuje vypínat a zapínat osvětlení v domě. Umožňuje stmívání světel podle nároků uživatele a nastavování předem definovaných světelných scén. Inteligentní instalace také umožňuje sledovat stavy osvětlení. Díky inteligentní elektroinstalaci je možné využít centrální monitoring, který umožňuje řídit a ovládat osvětlení budovy z libovolně zvoleného místa. Monitoring nám poskytne vizualizaci stavů, zda jsou světla rozsvícena nebo zhasnuta a není proto nutné budovu procházet a kontrolovat osvětlení. Inteligentní světelné předřadníky, které komunikují přes sběrnici DALI, umožňují zasílání informací, že je nutné vyměnit lampu nebo informaci o poruše komponenty. Pomocí hlášení těchto stavů, můžeme zvýšit spolehlivost systému, plánovat údržbu a předejít poškození předřadníku svítidla nebo dokonce vzniku požáru. Sledování stavů svítidel je vhodné sledovat i k rychlejší diagnostice závady a jejímu rychlému odstranění, není nutné obcházet všechna svítidla a zkoušet zda jsou funkční. Řízené ovládání osvětlení je vhodné využít i z ekonomického hlediska. Standartní elektroinstalace nám umožňuje jen zapnout nebo vypnout osvětlení a tak je spotřeba svítidel závislá na obsluze jak bude využívat osvětlení. Díky pokročilému řízení můžeme sledovat celkovou spotřebu osvětlení, případně upravit intenzitu osvětlení a tím snižovat náklady na spotřebovanou energii. [1] [25]

Pro sledování stavů a řízení osvětlení je nutné do projektu přidat datové body pro jednotlivé úkony. Adresa datového bodu je adresa, která nám určuje, jaká data se pro konkrétní stav budou z PLC získávat.

### **Adresa datového bodu pro stmívání svítidla**

test\Router 905 @ 32.201\DALI Subnet @ 32.201.1\HELVAR LC30-E-DA @ 32.201.1.1<output level>

### **Adresa datového bodu pro zobrazení výkonu svítidla**

test\Router 905 @ 32.201\DALI Subnet @ 32.201.1\HELVAR LC30-E-DA @ 32.201.1.1<load power>

### **Adresa datového bodu pro zobrazení přítomnosti předřadníku**

test\Router 905 @ 32.201\DALI Subnet @ 32.201.1\HELVAR LC30-E-DA @ 32.201.1.1<is missing>

### **Adresa datového bodu pro zobrazení poruchy svítidla**

test\Router 905 @ 32.201\DALI Subnet @ 32.201.1\HELVAR LC30-E-DA @ 32.201.1.1<lamp failure>

**Zdroj dat OPC** – OPC cache je pomocná paměť, do které jsou ukládána data z PLC. U OPC device jsou data získávána přímo z PLC, tento přístup k datům je pomalejší než získávání dat z OPC cache. (26)

**Moduly** - určují seznam modulů, do kterých se má signál posílat a zpracovávat.

**Příznaky pro historii** – určují, jak se signál bude vyhodnocovat a zapisovat do historie signálů.

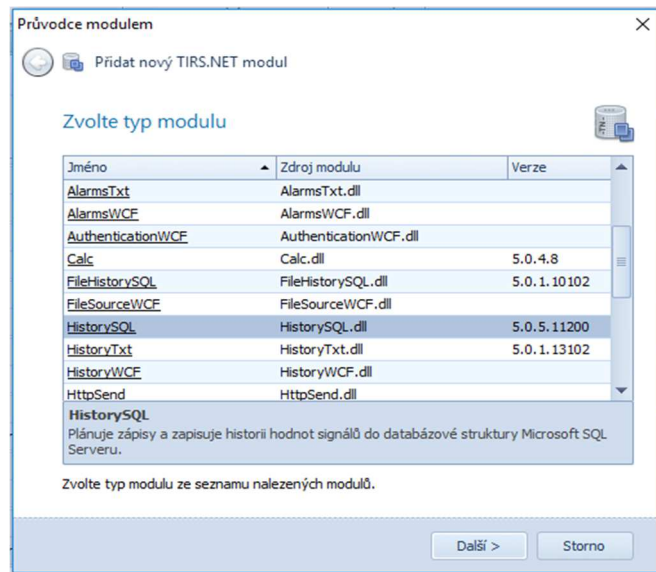
Pro sledování stavu svítidel zda svítí či nesvítí a výpočet celkové spotřeby osvětlení je použitý modul Calc, který bude popsán v následujících kapitolách s příloženými zdrojovými kódy.

## **3.6 Moduly**

Moduly slouží ke zpracování a ukládání dat ze signálů. V této bakalářské práci jsou použity moduly HistorySQL a modul Calc. Pomocí těchto modulů budeme moci ukládat a zpracovávat data ze signálů.

### 3.6.1 HistorySQL

Modul HistorySQL slouží k zápisu signálů do Microsoft SQL databáze. Standardně se databázové soubory ukládají do C:\TNPRJ\DB\. Pro přidání modulu HistorySQL přejdeme v TN Project Studio do části „Modules“ a v menu vybereme položku „Přidat modul“. Zobrazí se nám dialogové okno, ve kterém zvolíme HistorySQL.dll, stiskneme tlačítko „další“ a zvolíme jméno pro vybraný modul.

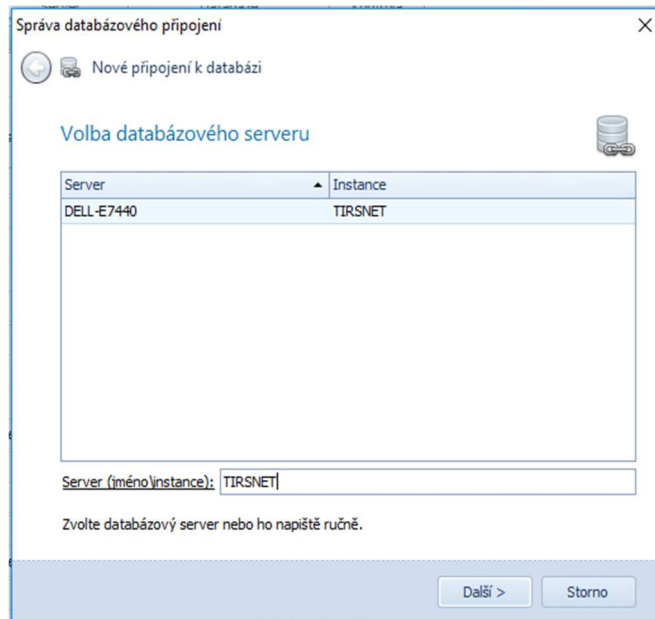


**Obrázek 20:** Modul HistorySQL

U takto vytvořeného modulu musíme přidat konfiguraci ve formátu XML. Konfigurační soubory se nachází ve složce C:\TNPRJ\Modules\_Config. Následně již můžeme modul povolit a použít v projektu. Pro použití modulu HistorySQL je nezbytné vytvoření databáze.

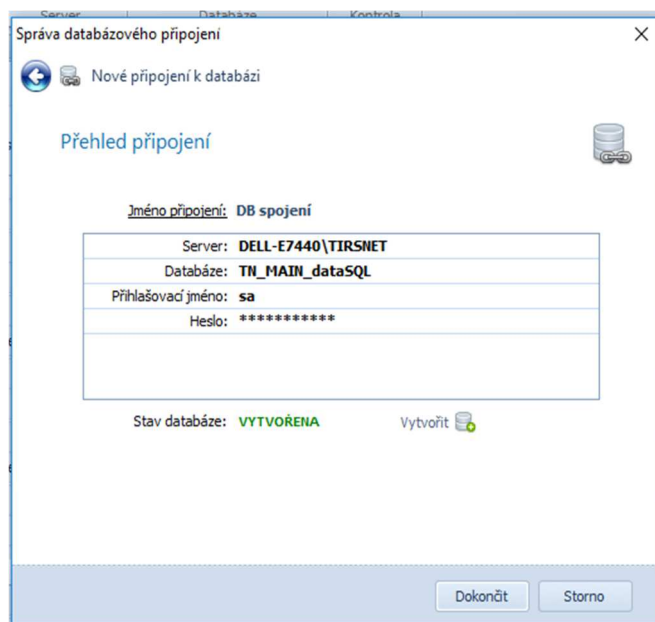
#### Vytvoření databáze

Pro vytvoření databáze vybereme v menu položku „Přidat připojení“. Následně vybereme server, který je nainstalován již z prvotní instalace systému TIRS.NET.



**Obrázek 21:** Vytvoření databáze

Následující dialogové okno nás vyzve k zadání přihlašovacích údajů, které je možné zvolit již při instalaci samotného systému. Po vložení přihlašovacích údajů následuje zvolení jména databáze a pojmenování tohoto spojení. V posledním kroku, který je vidět na obrázku 22 je nutné databázi vytvořit. Takto jsme vytvořili nové databázové připojení, které již můžeme využít v modulu HistorySQL. Pro každý projekt je vhodné vytvořit nové databázové připojení.

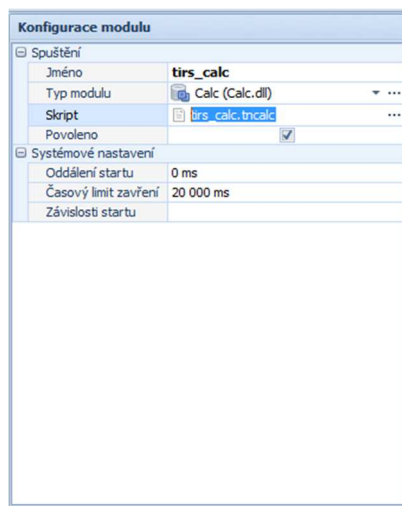


**Obrázek 22:** Nové připojení k databázi

### 3.6.2 Calc

Matematický modul Calc je možné rozšířit o programy definované uživatelem. Uživatelem napsané programy je možné naprogramovat v programovacím jazyce C# a Visual Basic. Calc tyto skripty zkompiluje při startu projektu. Matematický modul může být v jednom projektu spuštěn vícekrát. [27]

Modul Calc přidáme stejným způsobem jako modul HistorySQL. V průvodci novým modulem vybereme modul Calc a pokračujeme tlačítkem „další“, kde zvolíme jméno nového modulu a stiskneme dokončit. Když máme modul vytvořený, je potřeba ještě vytvořit soubor, do kterého se budou vkládat skripty. Tento soubor vytvoříme jako nový poznámkový blok s příponou .tncalc a vložíme do konfigurace modulu k položce skript.



Obrázek 23: Modul Calc

Modul Calc nám umožňuje zpracování signálů pomocí programovacího jazyka. Příložený **Zdrojový kód 1**, slouží pro výpočet celkové spotřeby dvou světél. Vypočítanou celkovou spotřebu světél program odešle do signálu LoadPower1, u kterého je nastavený parametr pro ukládání do databáze.

```
program name = "PrgPower" >
< signal > svetlo </ signal >
< signal > svetlo2 </ signal >
< code >
public object PrgPower(int iDuvod, SignalAttrib sa)
{
    double spotreba, spotreba2, svetlo2, report;
    spotreba = GetNumber(GetSignalAttrib("spotreba").tnValue, -1);
    svetlo2 = GetNumber(GetSignalAttrib("svetlo2").tnValue, -1);

    if (spotreba == -1)
```

```

{
    MonitorTextOut("chyba s nactenim hodnoty spotreba pro svetlo 1");
    return 1; // ukončení běhu
}
spotreba2 = (svetlo2 * 20) / 100;
double dLoadPower1 = spotreba + spotreba2;
MonitorTextOut("Celkova spotreba je: " + dLoadPower1 + " W");
WriteTirsSignal("LoadPower1", dLoadPower1);
WriteTirsSignal("spotreba2", spotreba2);
return 0;
}
</code>
</program>

```

### Zdrojový kód 1: Zdrojový kód pro výpočet celkové spotřeby

Níže uvedený **Zdrojový kód 2** slouží k vyhodnocování stavů svítidla, zda svítidlo svítí nebo nesvítí. Program výsledek zasílá do signálu sign1, který má nastavené parametry pro ukládání hodnot do databáze.

```

<program name = "PrgStav" language="c#">
<signal>svetlo</signal>

<code>
public object PrgStav(int iDuvod, SignalAttrib sa)
{
    int signal = 0;
    double svetlo = GetNumber(GetSignalAttrib("svetlo").tnValue, -1);

    if (svetlo == -1)
    {
        MonitorTextOut("chyba s nactenim hodnoty svetlo");
        return 1;
    }

    if (svetlo > 0)
    {
        signal = 1;
        MonitorTextOut("Svetlo sviti");
    }
    else {
        MonitorTextOut("Svetlo nesviti");
    }

    WriteTirsSignal("sign1", signal);

    return 0;
}
</code>
</program>

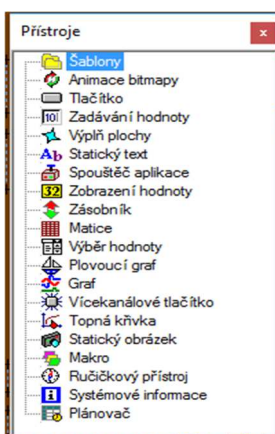
```

### Zdrojový kód 2: Zdrojový kód pro sledování stavu svítidla



### 3.6.3 TN Panel

TN panel slouží k vizualizaci technologií pomocí grafických schémat a přehledů. V tomto panelu je možné technologii monitorovat, ale i řídit. S TN panelem se pracuje v provozním režimu nebo v editačním režimu. Provozní režim umožňuje monitorovat a řídit technologie. Editační režim slouží k vytváření nebo editaci vizualizačních stránek. Jako hlavní nástroj pro vytvoření vizualizací slouží „Přístroje“, které otevřeme pomocí klávesové zkratky Ctrl+Shift+T. Pro přehled signálů slouží pomůcka „Signály“, kterou otevřeme pomocí klávesové zkratky Ctrl+Shift+S. [28]



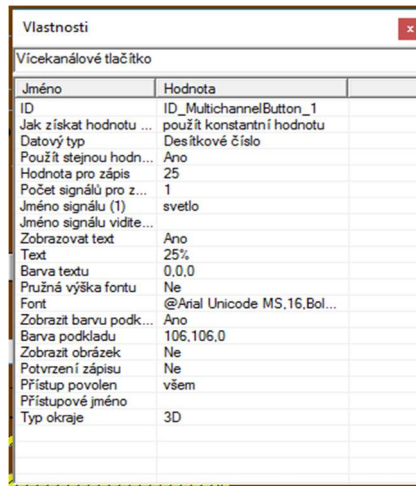
Obrázek 24: Přístroje v TN Panelu

#### Založení nového panelu

Pro založení nového vizualizačního panelu otevřeme aplikaci TN Panel a v menu vybereme možnost „soubor“ a jako další možnost „nový“, případně můžeme nový panel vytvořit pomocí klávesové zkratky Ctrl+N.

#### Tvorba vizualizačního panelu

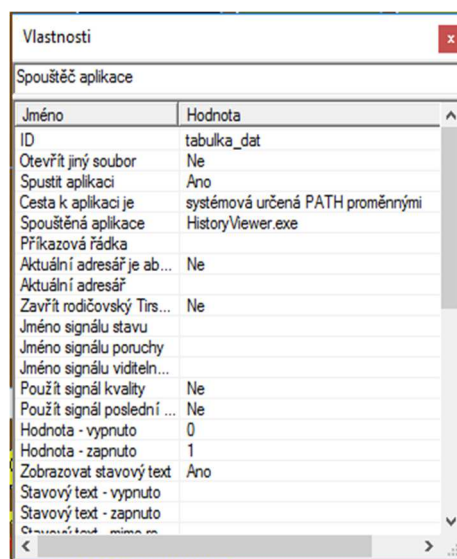
Pro tvorbu vizualizačního panelu použijeme pomůcku přístroje. Přístroje otevřeme z menu → pomůcky → přístroje nebo pomocí klávesové CTRL+SHIFT+T. Z přístrojů vybereme požadovanou možnost a přesuneme ji na plochu panelu. Když máme přidáný příslušný prvek, tak si zobrazíme jeho vlastnosti. Poté je možné nastavit požadované vlastnosti jako je signál, hodnota pro zápis, jméno signálu atd.



**Obrázek 25:** Vlastnosti tlačítka

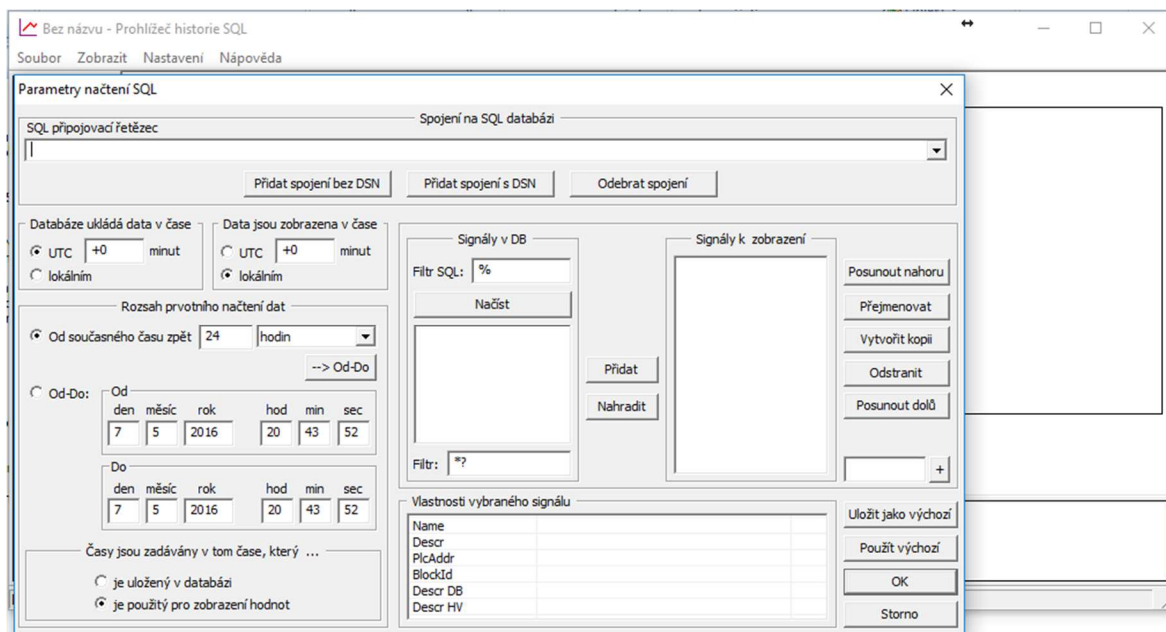
### Tvorba grafů a tabulek

Pro přidání grafů a tabulek vybereme z přístrojů možnost spouštěč aplikace, kterou přidáme do panelu a zobrazíme si jeho vlastnosti. Jako první parametr nastavíme cestu k aplikaci jako systémově určenou cestu a spouštěnou aplikaci HistoryViewer.exe, která se nachází v adresáři C:\Program Files (x86)\CORAL\TIRSNET\HistoryViewer.exe.



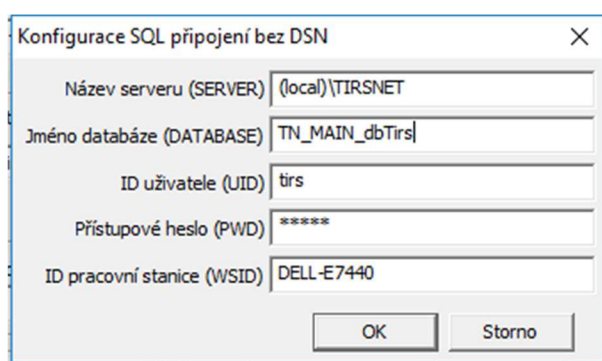
**Obrázek 26:** Vlastnosti spouštěné aplikace

Po vyplnění těchto dvou parametrů vlastnosti uzavřeme. Nyní spustíme spouštěč aplikace a v menu zvolíme Nastavení → DataSQL, poté se nám zobrazí okno pro konfiguraci připojení k databázi, do které se nám ukládají data ze signálů.



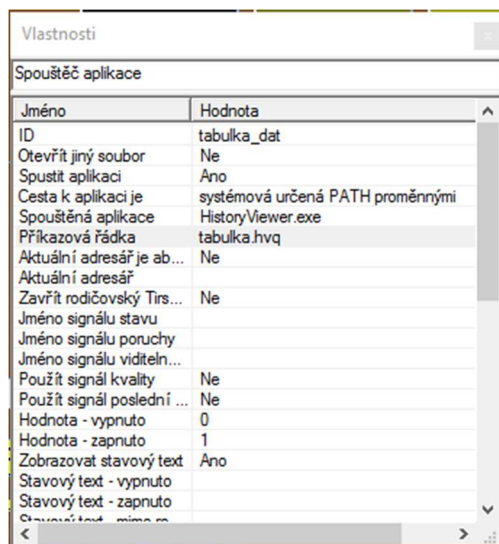
**Obrázek 27:** Spojení na SQL databázi

V části spojení na SQL databázi vybereme možnost „Přidat spojení bez DSN“. V konfiguraci vyplníme jméno databáze a přihlašovací údaje. Standardní jméno je tirs a standardní heslo coral.



**Obrázek 28:** Konfigurace SQL připojení

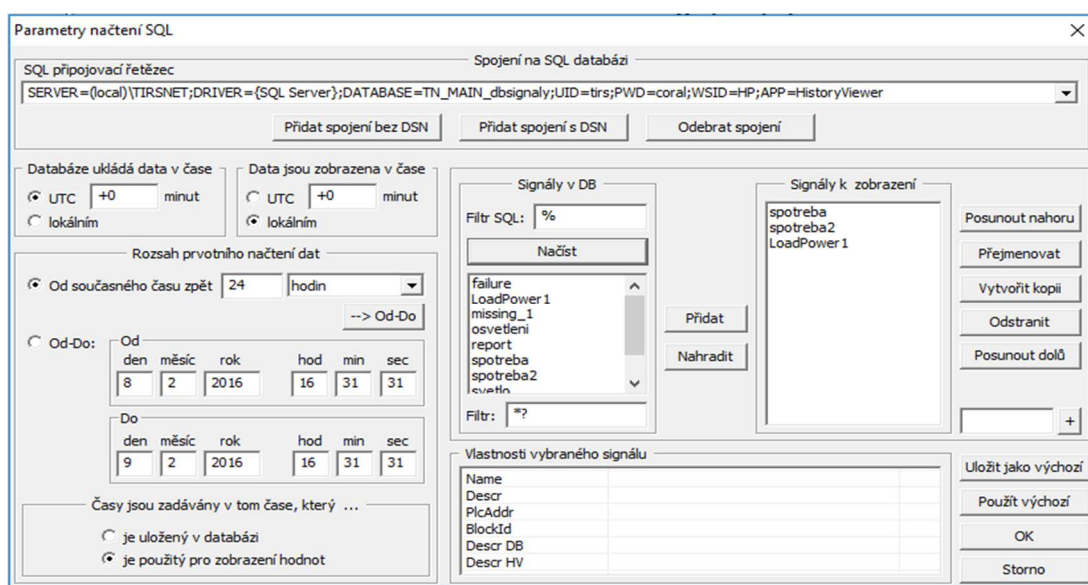
Po připojení k databázi načteme signály pomocí tlačítka „Načíst“ a můžeme přidat signály, které bude možné zobrazit v náhledu grafu nebo tabulky. Tuto volbu potvrdíme tlačítkem OK. Nyní jsme v prohlížeči historie SQL, kde můžeme zobrazit graf pomocí klávesové zkratky Ctrl+G nebo zobrazit tabulku pomocí klávesové zkratky Ctrl+T. Tabulku a graf uložíme z menu → Soubor → Uložit jako. Tabulku a graf uložíme s pojmenováním tabulka.hvq a graf.hvq. Nyní si otevřeme spouštěč aplikací a do příkazové řádky vyplníme pojmenované náhledy tabulky a grafu. Takto máme vytvořené spouštěče aplikací pro náhled grafů a tabulek.



Obrázek 29: Spouštěč aplikace - příkazová řádka

## Přidání signálů

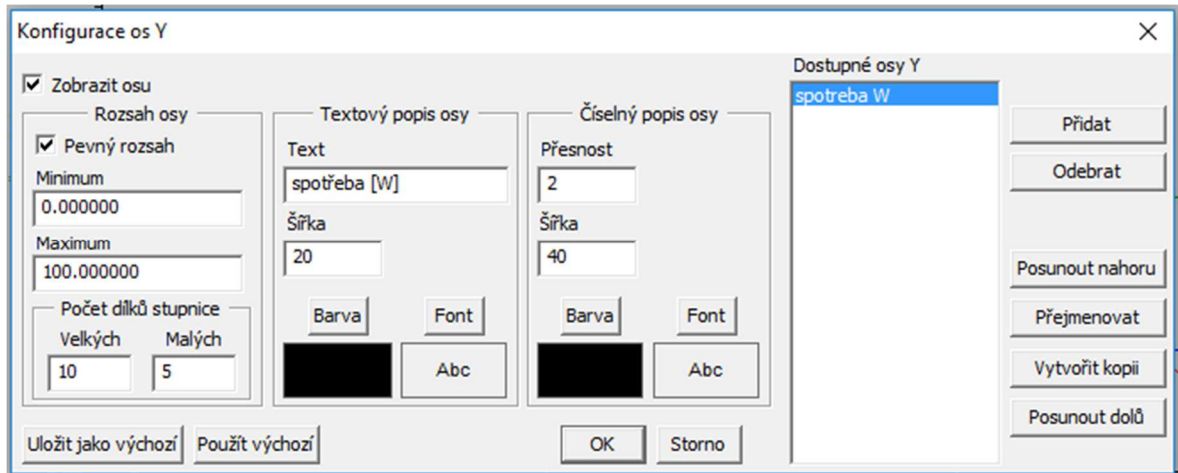
K zobrazení signálů v grafech a tabulkách je nutné tyto signály přidat k zobrazení. To uděláme tak, že si otevřeme graf.hvq, který jsme vytvořili v předchozím kroku. V menu zvolíme možnost nastavení → DataSQL. Zobrazí se nám dialogové okno pro načtení SQL. V části nazvané signály v DB stiskneme tlačítko „Načíst“ pro načtení signálů, které jsou uloženy v databázi. Nyní si můžeme vybírat a přidávat signály, které chceme zobrazit v pohledech.



Obrázek 30: Signály k zobrazení

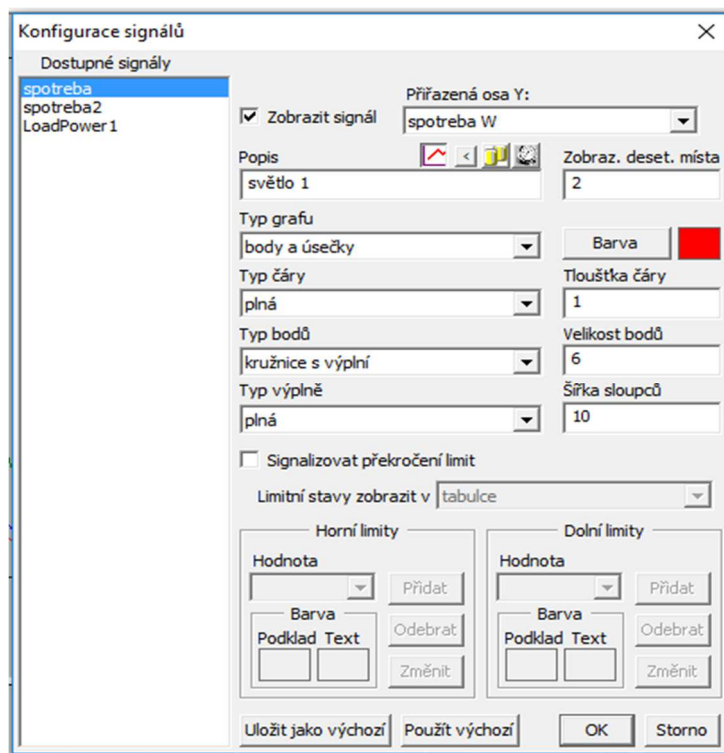
## Nastavení grafu

Pro nastavení grafu, je nutné ke každému signálu přiřadit osu Y. V prohlížeči historie SQL, kde máme otevřený graf.hvq vybereme z menu možnost Nastavení → Graf → Vlastnosti os Y. Přidáme novou osu a nastavíme rozsah této osy.



**Obrázek 31:** Konfigurace osy Y

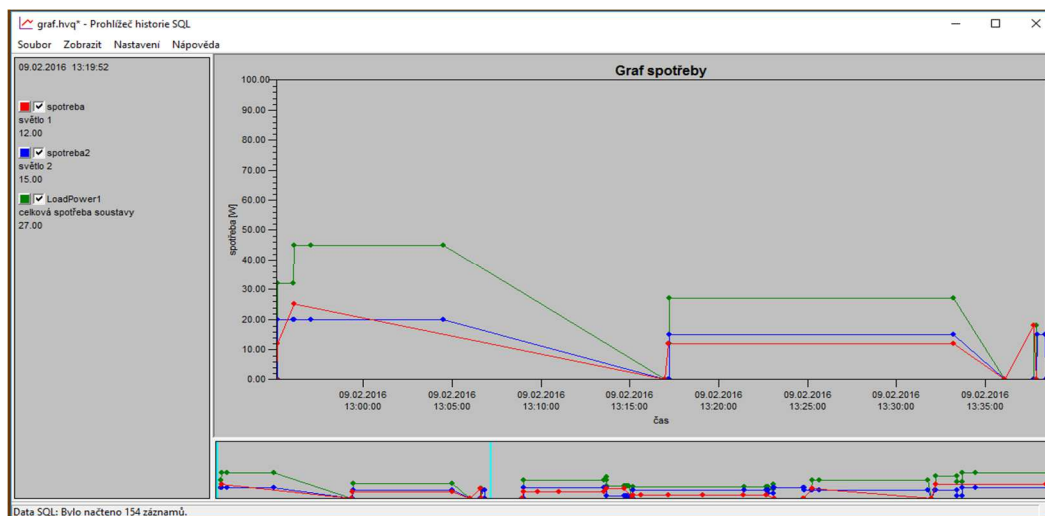
Když máme přidanou osu, můžeme k této ose přiřadit signály. Z menu vybereme Nastavení → Graf → Vlastnosti signálů. Zobrazí se nám dialogové okno konfigurace signálů. Z dostupných signálů vybereme jen ty signály, které chceme zobrazit v grafu. Ke každému signálu přiřadíme osu Y a zaškrtneme položku zobrazit signál. U signálu dále můžeme změnit barvu signálu, styl čáry nebo nastavit limity pro překročení hodnot.



**Obrázek 32:** Konfigurace signálů

Takto jsme dokončili nastavení grafu. Pro zachování nastavení grafu, musíme konfiguraci uložit. V menu vybereme možnost Soubor → Uložit nebo pomocí klávesové zkratky Ctrl+S.

Na obrázku 33 je zobrazen pro příklad graf z prohlížeče událostí v náhodném časovém okamžiku. Graf zobrazuje průběh spotřeby dvou svítidel (světlo1, světlo2) a celkovou spotřebu, která je daná součtem těchto dvou spotřeb svítidel.



**Obrázek 33:** Graf spotřeby

## **Nastavení tabulky**

Prohlížeč historie SQL podporuje režim zobrazení grafu nebo zobrazení tabulky. Signály k zobrazení přidáme stejným způsobem jako v odstavci Přidání signálů. Po přidání signálů zvolíme v menu možnost Zobrazit → Tabulka nebo použijeme klávesovou zkratku Ctrl+T.

Na obrázku 34 je pro příklad zobrazena tabulka dat s hodnotami signálů z náhodně vybraného časového úseku ze systému TIRS.NET.

**Missing\_1** – Zobrazuje stav, zda je připojený LED driver a nabývá hodnoty 0 pro připojený předřadník a hodnoty 1 pro odpojený předřadník.

**Failure** – Zobrazuje stav selhání lampy, nabývá hodnoty 0 pro stav v pořádku a hodnoty 1 pro stav selhání lampy.

**Světlo** – Hodnota světlo nám udává procentuální rozsvícení světla jednoho konkrétního světla1, nabývá hodnoty 0-100 %.

**Spotřeba** – Hodnota spotřeba nám udává, spotřebu světla1.

**Světlo2** - Hodnota světlo2 nám udává procentuální rozsvícení světla jednoho konkrétního světla2, nabývá hodnoty 0-100 %.

**Spotřeba** – Hodnota spotřeba nám udává, spotřebu světla2.

**LoadPower1** – Hodnota LoadPower1 nám udává celkovou spotřebu světel, která je daná součtem světla1 a světla2.

**Sign1**- Hodnota Sign1 nám udává, zda je světlo rozsvícené nebo zhasnuté, nabývá dvou hodnot, pro stav rozsvíceno nabývá hodnoty 1 a pro stav zhasnuto nabývá hodnoty 0.

tabulka\_dat.hvq - Prohlížeč historie SQL

Soubor Zobrazit Nastavení nápověda

Datum	missing_1 /	failure /	svetlo /	spotreba /	svetlo2 /	spotreba2 /	LoadPower1 /	sign1 /
15.01.2016 12:33:08			50.00	12.00		0.00	12.00	
15.01.2016 12:34:00	0.00	0.00						
15.01.2016 12:34:44				12.00				
15.01.2016 12:34:46			50.00		0.00	0.00	12.00	1.00
15.01.2016 12:35:00	0.00	0.00						
15.01.2016 12:36:00	0.00	0.00						
15.01.2016 12:37:00	0.00	0.00						
15.01.2016 12:38:00	0.00	0.00						
15.01.2016 12:38:01				12.00				
15.01.2016 12:38:03			50.00		0.00	0.00	12.00	1.00
15.01.2016 12:38:21			50.00			0.00	12.00	1.00
15.01.2016 12:38:22			0.00	0.00		0.00	0.00	0.00
15.01.2016 12:38:56			0.00			0.00	0.00	0.00
15.01.2016 12:38:57			50.00	12.00		0.00	12.00	1.00
15.01.2016 12:39:00	0.00	0.00						
15.01.2016 12:39:51			0.00	0.00		0.00	0.00	0.00
15.01.2016 12:40:00	0.00	0.00						
15.01.2016 12:41:00	0.00	0.00						

Data SQL: Bylo načteno 1264 záznamů.

Obrázek 34: Tabulka dat



## **4 Zpřístupnění dat**

Exporty dat jsou nedílnou součástí analýzy systému. Díky těmto datům můžeme sledovat systém, vyhodnocovat poruchovost vybraných komponent. V případě vyšší poruchovosti můžeme vyměnit součástku nebo svítidlo za jiný typ a tím si například zvyšovat spolehlivost systému. Pomocí analýzy dat můžeme plánovat pravidelné kontroly nebo i sledovat jak často konkrétní světlo svítilo.

### **4.1 Zpřístupnění dat pomocí www**

Systém TIRS.NET umožňuje publikovat vizualizaci technologie na web pomocí nadstavbového nástroje TirsPanelWeb. Pro vizualizaci je nutné, aby na serveru, kde je instalovaný systém TIRS.NET, běžel i webový server. Pomocí síťového připojení je možné sledovat stavy různých technologií, jak v interní síti, tak i v internetu. Tato nadstavba je vhodná pro občasné sledování stavů bez možnosti zadávat příkazy. Nadstavba TirsPanelWeb nedokáže publikovat grafy ani práci s historickými daty jako jsou tabulky. Systém TIRS.NET verze 5, který je v této práci použitý neumožňuje pomocí nadstavby zadávat příkazy, pracovat s grafy a tabulkami hodnot. Systém TIRS.NET od verze 6 již umožňuje pomocí nadstavby pracovat s grafy i zadávat příkazy pro danou technologii, ale stále nedokáže pracovat s tabulkami hodnot. [29]

### **4.2 Zpřístupnění dat pomocí tisku**

Prohlížeč historie SQL načítá data z databáze SQL. Pomocí tohoto nástroje jsou vytvořeny grafy a tabulky hodnot. Prohlížeč historie SQL umožňuje export tabulky do souboru CSV, který je možné následně zpracovat v tabulkovém editoru. Tento export je možné provést pravým kliknutím myši na tabulku a zvolit možnost „uložit tabulku“, ale je také možné tabulku zkopírovat a vložit do textového editoru. Jako další možnosti zpracování umožňuje tabulky a grafy vytisknout. Pro vytisknutí grafů a tabulek v prohlížeči historie SQL v menu vybereme položku soubor → tisk. [30]

## Závěr

Tato bakalářská práce byla zaměřena na sledování stavů osvětlení. Pro řízení a sledování osvětlení je použitý systém TIRS.NET, který umožňuje komunikovat s hardwarem přes OPC protokol. Pro osvětlovací systém byl použit hardware od firmy Helvar, konkrétně router Imagine 920 a LED driver LC1x30-E-DA.

V práci je uveden popis osvětlovacího systému a řídicí systém TIRS.NET. V další části této práce jsou popsány komunikační protokoly a postup od tvorby aplikace pro sledování stavu osvětlení až po ukládání do databáze.

V současné době tento projekt není aplikován na žádné budově. Instalace systému do budovy není v současné chvíli plánována z několika důvodů.

Prvním důvodem jsou problémy v komunikaci pomocí protokolu OPC, který má za následek výpadky komunikace mezi obslužným serverem a PLC. Vzhledem k tomu, že systém TIRS.NET komunikuje s routerem pomocí sítě ethernet, byl vyzkoušen jiný ethernetový kabel. K eliminaci problému byl použit další počítač s rozdílnou síťovou kartou a různou verzí operačního systému, ale problém s komunikací nadále přetrvával. Dále jsem vyzkoušel v aplikaci TN Project zakázat veškeré moduly a signály, kde byly následně povolovány a testovány, ale problém v komunikaci stále přetrvával. Důvodem bylo eliminovat špatnou konfiguraci modulů a signálů, které by mohly způsobovat výpadky komunikace. V současné době neexistuje v systému TIRS.NET jiný komunikační konektor, který by byl schopen komunikovat se zařízením od firmy Helvar. Firma Coral s.r.o., která je výrobcem softwaru TIRS.NET, umožňuje vytvořit komunikační konektor pro méně časté technologie na zakázku.

Druhým důvodem jsou problémy s aplikací Helvar Designer, kterou je nutné spustit a ručně připojit router, jinak by aplikace systému TIRS.NET nekomunikovala s tímto routerem. Tento problém by mohl nastat při výpadku napájení, při němž by systém ztratil komunikaci s routerem.

Třetím důvodem jsou problémy s publikací vizualizací na web s grafy a tabulkami. Systém TIRS.NET v současné době neumí zpracovat grafy a tabulky pomocí nástroje TirsPanelWeb, který slouží k publikování vizualizace technologie na webové stránky.

Pro nasazení aplikace by bylo vhodné naprogramovat komunikační konektor, který by nahradil současně použitý OPC konektor a umožňoval by komunikaci přímo s hardwarem. Dále by bylo vhodné vyřešit navázání komunikace s routerem bez aplikace Helvar designer. Problém s aplikací Helvar designér by vyřešil komunikační konektor, který by nahradil konektor TnOPC a komunikoval by přímo s připojeným zařízením od firmy Helvar.

## Seznam použité literatury

- [1] ŽÁK, Petr. Energetická náročnost a úspory u osvětlovacích soustav. *Tzbinfo.cz* [online]. Praha: leden 26, 2015 [cit. 2016-07-18]. Dostupné z: <http://elektro.tzb-info.cz/osvetleni/12254-energeticka-narocnost-a-uspory-u-osvetlovacich-soustav>
- [2] Osvětlovací soustavy. *Fa.cvut.cz* [online]. [cit. 2016-06-16]. Dostupné z: <https://www.fa.cvut.cz/attachments/BAhbBlSHOgZmSSIdNTQ4YWRiZDU1MDE2NTMwNDkyMDAzMDkyBjoGRVQ/OSV%C4%9ATLOVAC%C3%8D%20SOUSTAVY%20-%20ateli%C3%A9r.pdf?sha=9cfd4aca>
- [3] Osvětlovací systémy. *Docplayer.cz* [online]. [cit. 2016-06-16]. Dostupné z: <http://docplayer.cz/19206741-Osvetlovaci-systemy-p-e-o-p-l-e-i-n-o-v-a-t-i-o-n-s-s-o-l-u-t-i-o-n-s.html>
- [4] Imagine router (920). *Helvar.com* [online]. March 03, 2016 [cit. 2016-06-16]. Dostupné z: <http://www.helvar.nl/files/downloads/920imagerouterdatasheetiss07uk.pdf>
- [5] Led komponenty. *Dna.cz* [online]. Srpen, 2012 [cit. 2016-07-18]. Dostupné z: <http://www.dna.cz/download/2012-led-komponenty.pdf>
- [6] LC1x30-E-DA. *Helvar.com* [online]. April 17, 2015 [cit. 2016-07-18]. Dostupné z: [http://www.helvar.com/sites/default/files/product\\_datasheets/LC1x30-E-DA\\_datasheet\\_REF\\_T220041G\\_27042015\\_EN.pdf](http://www.helvar.com/sites/default/files/product_datasheets/LC1x30-E-DA_datasheet_REF_T220041G_27042015_EN.pdf)
- [7] LC1x30-E-DA. *Helvar.com* [online]. April 17, 2015 [cit. 2016-07-18]. Dostupné z: [http://www.helvar.com/sites/default/files/product\\_datasheets/LC1x30-E-DA\\_datasheet\\_REF\\_T220041G\\_27042015\\_EN.pdf](http://www.helvar.com/sites/default/files/product_datasheets/LC1x30-E-DA_datasheet_REF_T220041G_27042015_EN.pdf)
- [8] VAŇUŠ, Jan. Systémová technika budov a bytů. *Feil.vsb.cz* [online]. Leden, 2003 [cit. 2016-07-18]. Dostupné z: <http://feil.vsb.cz/kat420/vyuka/TZB/systemova%20technika%20budov.pdf>
- [9] HALUZA, Miroslav. MACHÁČEK, Jan. Klasická versus inteligentní elektroinstalace. *Tzbinfo.cz* [online]. Zář 19, 2011 [cit. 2016-08-09]. Dostupné z: <http://elektro.tzb-info.cz/domovni-elektroinstalace/7842-klasicka-versus-inteligentni-elektroinstalace>

- [10] MLČÁK, Tomáš. Systémová technika budov / způsoby a principy realizace. *Fei1.vsb.cz* [online]. Březen, 2006 [cit. 2016-08-10]. Dostupné z: [http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/1\\_klasicka\\_elektroinstalace\\_a\\_systemova\\_tehnika\\_budov.pdf](http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/1_klasicka_elektroinstalace_a_systemova_tehnika_budov.pdf)
- [11] NEVAŘILOVÁ, Ludmila. Druhy a typy sběrníkových systémů. *Uvp3d.cz* [online]. [cit. 2016-08-10]. Dostupné z: [http://uvp3d.cz/dum/?page\\_id=2884](http://uvp3d.cz/dum/?page_id=2884)
- [12] TIRS.NET – architektura systému. *Tirs.cz* [online]. [cit. 2016-08-10]. Dostupné z: <http://www.tirs.cz/scadahmi-systemy-tirs/tirs-net-architektura-systemu/>
- [13] Co znamená SCADA/HMI? *Reliance.cz* [online]. [cit. 2016-08-10]. Dostupné z: <https://www.reliance.cz/cs/products/what-does-scada-hmi-mean>
- [14] BALDA, Pavel. SCADA a HMI systémy. *Vendulka.zcu.cz* [online]. [cit. 2016-08-10]. Dostupné z: [http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08\\_SCADA\\_HMI.pdf](http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08_SCADA_HMI.pdf).
- [15] What is SCADA? *Inductiveautomation.com* [online]. [cit. 2016-08-10]. Dostupné z: <https://inductiveautomation.com/what-is-scada>
- [16] Univerzální data – management pro výrobní stroje CNC. *Homel.vsb.cz* [online]. [cit. 2016-08-10]. Dostupné z: <http://homel.vsb.cz/~los032/Prostredky%20automatickeho%20rizeni/Preklad/preklad%20CZ.pdf>
- [17] BALÁŠ, Jaroslav. Klientské rozhraní aplikací SCADA. *Dip.felk.cvut.cz* [online]. [cit. 2016-08-10]. Dostupný z: [https://dip.felk.cvut.cz/browse/pdfcache/balasz2\\_2007dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/balasz2_2007dipl.pdf)
- [18] Discover DALI – excellent system performance. *Dali-ag.org* [online]. [cit. 2016-08-10]. Dostupné z: <http://www.dali-ag.org/discover-dali.html>
- [19] Elektronické předřadníky + stmívání. *Elkovo-cepelik.cz* [online]. [cit. 2016-08-10]. Dostupné z: <http://www.elkovo-cepelik.cz/editor/filestore/File/katalogove-listy-2014/elkovo-dimming.pdf>
- [20] Český výrobce svítidel. *Modus.cz* [online]. [cit. 2016-07-05]. Dostupné z: <http://www.modus.cz/cze/getfile.php?FileID=6282>

- [21] ŠIMMER, Jakub. Řízení a monitoring systému osvětlení. *CVUT.cz* [online]. Praha: 2014 [cit. 2016-07-05]. Dostupné z: [https://support.dce.felk.cvut.cz/mediawiki/images/f/f3/Bp\\_2014\\_simmer\\_jakub.pdf](https://support.dce.felk.cvut.cz/mediawiki/images/f/f3/Bp_2014_simmer_jakub.pdf)
- [22] DALI – profesionální rozhraní pro všechny komponenty osvětlení. *Osram.cz* [online]. [cit. 2016-07-05]. Dostupné z: [http://www.osram.cz/osram\\_cz/novinky-a-znalosti/systemy-rizeni-osvetleni/technologie/dali/index.jsp](http://www.osram.cz/osram_cz/novinky-a-znalosti/systemy-rizeni-osvetleni/technologie/dali/index.jsp)
- [23] Co je OPC? OPC server? OPC klient? *Foxon.cz* [online]. [cit. 2016-07-05]. Dostupné z: <http://www.foxon.cz/cs/blogs/80-co-je-opc-opc-server-opc-klient-.html>
- [24] *Konfigurace komunikačních konektorů*. CORAL s.r.o.: 2013, Hradec Králové. Elektronická dokumentace k software TIRS.NET.
- [25] Úspory díky řízení osvětlení. *Feramat.com* [online]. [cit. 2016-08-10]. Dostupné z: <http://www.feramat.com/uspory-osvetleni.html>
- [26] Komunikace přes standardní rozhraní OPC. *Promotic.eu* [online]. [cit. 2016-08-10]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPC.htm>
- [27] *Calc.dll – Matematický modul (modul pro matematické a jiné programové operace definované uživatelem)*. CORAL s.r.o.: 2012, Hradec Králové. Elektronická dokumentace k software TIRS.NET.
- [28] *TN Panel – Návrhář vizualizace*. CORAL s.r.o.: 2009, Hradec Králové. Elektronická dokumentace k software TIRS.NET.
- [29] *TirsPanelWeb (nastavba systému TIRS.NET pro prohlížení vizualizací pomocí webové technologie)*. CORAL s.r.o.: 2011, Hradec Králové. Elektronická dokumentace k software TIRS.NET.
- [30] *HistoryViewer – Prohlížeč historie*. CORAL s.r.o.: 2011, Hradec Králové. Elektronická dokumentace k software TIRS.NET.
- [31] *Instalace systému TIRS.NET*. CORAL s.r.o.: 2012, Hradec Králové. Elektronická dokumentace k software TIRS.NET.

## **Přílohy**

### **A Obsah přiloženého CD**

- Text bakalářské práce
  - Bakalarska\_prace\_2016\_Jan\_Horacek.doc
  - Bakalarska\_prace\_2016\_Jan\_Horacek.pdf
  - Kopie\_zadani\_bakalarska\_prace\_2015\_Jan\_Horacek.pdf
- Fotografie
  - snímky vizualizační obrazovky
- Aplikace
  - Aplikace pro sledování stav osvětlení