



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

VZDÁLENÉ ŘÍZENÍ A VIZUALIZACE DEMONSTRAČNÍHO PANELU KNX

REMOTE CONTROL AND VISUALISATION OF A KNX DEMONSTRATION BOARD

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Luboš Kelnar

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Petr Fiedler, Ph.D.

BRNO 2025



Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Luboš Kelnar

ID: 221302

Ročník: 3

Akademický rok: 2024/25

NÁZEV TÉMATU:

Vzdálené řízení a vizualizace demonstračního panelu KNX

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat vzdálené řízení a vizualizaci demonstračního panelu KNX pro ovládání funkcí osvětlení, žaluzií, topení a klimatizace. Vzdálené řízení bude realizováno prostřednictvím PLC a rozhraním IP/KNX. Vizualizace bude zobrazovat aktuální stav panelu a řešena bude přes webovou aplikaci navrženou pro mobilní zařízení/ tablet. Součástí řešení je i vytvoření dynamických světlených scén, kde bude panel automaticky regulovat osvětlení, dle připravených scénářů.

- 1) Seznamte se s technologií KNX
- 2) Prozkoumejte možnosti řízení systému KNX pomocí vizualizační platformy
- 3) Navrhněte řízení KNX panelu prostřednictvím zvolené vizualizace
- 4) Vytvořte vizualizaci řízení panelu pro mobilní zařízení/tablet pro účel prezentace funkcí
- 5) Navrhněte a ověřte dynamické světelné scény

DOPORUČENÁ LITERATURA:

[1] Dokumentace k technologii KNX

[2] Dokumentace k PLC Tecomat

Termín zadání: 10.2.2025

Termín odevzdání: 28.5.2025

Vedoucí práce: doc. Ing. Petr Fiedler, Ph.D.

Konzultant: Ing. Branislav Bátora, Ph.D.

Ing. Miroslav Jirgl, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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.

ABSTRAKT

Cílem této bakalářské práce je realizace vzdáleného řízení a vizualizace demonstračního panelu KNX pomocí programovatelného automatu společnosti TECO a moderních zobrazovacích nástrojů s využitím Dockeru. Vizualizace je dostupná prostřednictvím webové aplikace optimalizované pro mobilní zařízení a tablety, s přehledným menu rozděleným do sekcí. Práce nejprve seznamuje s technologií KNX a možnostmi řízení systému, dále popisuje tvorbu řídicí logiky v softwaru ETS a možnosti vizualizace pomocí webového serveru automatu. Závěrem jsou diskutována alternativní řešení s využitím Raspberry Pi a Docker kontejnerů. Výsledkem je návrh a implementace flexibilního systému pro vzdálené ovládání a monitorování KNX instalací.

KLÍČOVÁ SLOVA

KNX, ETS, MQTT, Docker, vizualizace, inteligentní elektroinstalace

ABSTRACT

The aim of this bachelor thesis is to implement remote control and visualization of a KNX demonstration panel using a TECO Programmable Logic Controller and modern display tools using Docker. The visualization is accessible through a web application optimized for mobile devices and tablets, with a clear menu divided into sections. The paper first introduces KNX technology and the possibilities of controlling the system, then describes the creation of control logic in ETS software and the possibilities of visualization using the automaton web server. Finally, alternative solutions using Raspberry Pi and Docker containers are discussed. The result is the design and implementation of a flexible system for remote control and monitoring of KNX installations.

KEYWORDS

KNX, ETS, MQTT, Docker, visualization, intelligent wiring

KELNAR, Luboš. *Vzdálené řízení a vizualizace demonstrativního panelu KNX*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2025. Vedoucí práce: doc. Ing. Petr Fiedler, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Luboš Kelnar
VUT ID autora:	221 302
Typ práce:	Bakalářská práce
Akademický rok:	2024/25
Téma závěrečné práce:	Vzdálené řízení a vizualizace demonstrativního panelu KNX

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Petru Fiedlerovi, Ph.D a konzultantu panu Ing. Branislavu Bátorovi Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	23
Cíle práce	25
1 Sběrnicový systém KNX	27
1.1 Historie	27
1.1.1 EIBA	27
1.1.2 KNX	28
1.2 Možnosti použití technologie	28
1.3 Sběrnicové instalace	29
1.3.1 Sběrnicové přístroje	29
1.3.2 Adresování	30
1.3.3 Komunikace po sběrnici	31
1.3.4 KNXnet/IP	32
1.3.5 Datový bod	33
1.4 Zabezpečení	35
1.5 Topologie	36
1.5.1 Individuální adresa	37
1.5.2 Spojka	37
1.5.3 Routingové číslo	37
1.5.4 Interní a externí rozhraní	38
2 ETS	39
2.1 Tvorba instalace	39
2.2 Parametrisace tlačítek a detektoru pohybu	41
2.2.1 ABB - SBR/U6.0.1-84	42
2.2.2 Berker - 75663593	42
2.2.3 Ekinex - EK-ED2-TP-RW	43
2.2.4 Basalte - Senido 202-03	44
2.2.5 Simon - 8400100-039	44
2.2.6 HDL - M/TBP6.1-A2	45
2.2.7 Siemens - QMX3.P37	47
2.2.8 B.E.G - Indor 140-L-KNX-DX	48
2.3 Parametrisace akčních členů	48
2.3.1 ABB SA/S8.10.2.1	48
2.3.2 ABB - JRA/S4.230.2.1	49
2.3.3 HDL - M/R8.10.1	49

2.3.4	HDL - M/DRGBW4.1	50
2.4	Připojená komunikační rozhraní	50
2.5	Vytvoření skupinových adres projektu	53
3	Vizualizace skrze programovatelný logický automat	57
3.1	CP - 2007	57
3.2	Mosaic	58
3.2.1	Ovládací prvky	59
3.2.2	Komunikace KNX/IP	66
3.2.3	Komunikace MQTT	68
3.2.4	Vizualizace	72
4	Open-source platformy	73
4.1	Raspberry Pi 5	73
4.2	Docker Compose	74
4.2.1	Kontejnerizace	74
4.2.2	Tvorba YAML souboru	75
4.2.3	Portainer	76
4.3	Mosquitto	77
4.4	Home Assistant	77
4.5	InfluxDB	80
4.6	Grafana	80
Závěr		83
Literatura		85
Seznam symbolů a zkratek		91
Seznam příloh		93
A Skupinové adresy		95
B Globální proměnné a struktury PLC		101
C Definice funkčních složitějších funkčních bloků		103
C.0.1	fbRoomTempMod	103
C.0.2	Popis funkčního bloku fbBath	108
C.0.3	fbKitch	109
C.0.4	fbLivRoom	111
C.0.5	fbOutz	114

D Program komunikace mezi PLC a KNX	115
E Program komunikace mezi PLC a Home Assistantem	119
F Vizualizace pomocí WebMakeru	125
G Docker YAML soubory	129
G.0.1 Portainer YAML	129
G.0.2 Stack YAML	130
H Home Assistant - konfigurační soubor	133
I Nastavení přehledů v Grafaně	135
I.0.1 Grafana - Spotřeba	135
I.0.2 Grafana - Spínání	138
I.0.3 Grafana - Celkové měsíční náklady	140
I.0.4 Grafana - Průběh teploty	142

Seznam obrázků

1.1	Součásti sběrnicového přístroje [4]	29
1.2	Struktura individuální adresy [?]	30
1.3	Příklad struktury skupinových adres [6]	31
1.4	Struktura telegramu [6]	31
1.5	Struktura bitu kroucené dvojlinky [6]	32
1.6	Tunelování [12]	33
1.7	Routování [13]	34
1.8	Ukázka topologie KNX[8]	36
2.1	Projekt budovy v ETS	40
2.2	Pracovní plocha v ETS	41
2.3	Grafický návrh panelu [9]	41
2.4	Šestinásobné tlačítko ABB - SBR/U6.0.1-84 [14]	42
2.5	Osmínásobné tlačítko Berker - 75663593 [19]	43
2.6	Čtyřnásobné tlačítko Ekinex - EK-ED2-TP-RW [9]	43
2.7	Čtyřnásobné dotykové tlačítko Basalte - Senido 202-03 [17]	44
2.8	Čtyřnásobné tlačítko Simon - 8400100-039 [26]	45
2.9	Šestinásobné tlačítko HDL - M/TBP6.1-A2 [21]	45
2.10	Parametry scény A tlačítka HDL - M/TBP6.1-A2	46
2.11	Parametry scény B tlačítka HDL - M/TBP6.1-A2	46
2.12	Ovladací panel Siemens - QMX3.P37 [24]	47
2.13	Detektor pohybu B.E.G - Indor 140-L-KNX-DX [9]	48
2.14	Osmikanálový spínací člen ABB - SA/S8.10.2.1 [15]	49
2.15	Čtyřkanálový žaluziový člen ABB - JRA/S4.230.2.1 [16]	49
2.16	Osmikanálový spínací člen HDL - M/R8.10.1 [22]	50
2.17	Čtyřnásobný stmívací člen HDL - M/DRGBW4.1 [23]	50
2.18	IP Secure router Siemens - 5WG1 146-1AB03 [25]	51
2.19	Komunikační rozhraní Weinzierl - KNX IP BAOS 774 [28]	51
3.1	CP-2007 [29]	57
3.2	fbKNXVisuBool	60
3.3	fbKNXShutters	61
3.4	Průběh funkcí pro simulaci	63
3.5	Simulace teploty v místnosti	65
3.6	Ikony použité pro vizualizaci	72
4.1	Raspberry Pi5 [39]	74
4.2	Docker Stack	75
4.3	Portainer Stack	77
4.4	Home Assistant hrubá vizualizace	78

4.5	Home Assistant funkční vizualizace	79
4.6	Home Assistant vizualizace dle umístění	79
C.1	Logika funkčního bloku fbBath	108
C.2	Logika funkčního bloku fbKitch	110
C.3	Logika funkčního bloku fbLivRoom	113
C.4	Logika funkčního bloku fbOutz	114
F.1	WebMaker - přehled	125
F.2	WebMaker - Veranda	125
F.3	WebMaker - Kuchyně	126
F.4	WebMaker - Koupelna	126
F.5	WebMaker - Obývací pokoj	127
I.1	Vzhled přehledu v Grafaně	143

Seznam tabulek

1.1	Přehled nejpoužívanějších datových typů	34
1.2	Individuální adresy v topologii [8]	37
2.1	Datapointy pro komunikaci KNX IP BAOS 774 s PLC	52
2.2	Funkce kuchyňských tlačítek pro krátké stisknutí	53
2.3	Funkce kuchyňských tlačítek při dlouhém stisknutí	53
2.4	Funkce tlačítek obývacího pokoje při krátké stisknutí	54
2.5	Funkce tlačítek obývacího pokoje při dlouhém stisknutí	54
2.6	Funkce dotykových tlačítek při krátké stisknutí	55
2.7	Funkce dotykových tlačítek při dlouhém stisknutí	55
3.1	Definice KNX datapointů v PLC	66

Seznam výpisů

3.1	Definice funkčního bloku fbKNXVisuBool	59
3.2	Definice funkčního bloku fbKNXShutters	60
3.3	Implementace komunikace KNX/IP	67
3.4	Příklad zprávy v JSON	69
3.5	Příklad nastavení komunikace pro publikování zprávy	70
4.1	YAML soubor	75
B.1	Globální proměnné a struktury PLC	101
C.1	fbRoomTempMod	103
C.2	fbBath	108
C.3	fbKitch	109
C.4	fbLivRoom	111
C.5	fbOutz	114
D.1	Program komunikace mezi PLC a KNX	115
E.1	Program komunikace mezi PLC a Home Assistant – MQTT	119
G.1	Portainer YAML	129
G.2	Stack YAML	130
H.1	Home Assistant configuration.yaml	133
I.1	Grafana - Spotřeba klimatizace	135
I.2	Grafana - Spotřeba topení	135
I.3	Grafana - Spotřeba světla	136
I.4	Grafana - Rozložení spotřeby	136
I.5	Grafana - Spínání klimatizace	138
I.6	Grafana - Spínání topení	138
I.7	Grafana - Spínání světel	138
I.8	Grafana - Spínání žaluzií	139
I.9	Grafana - Celkové měsíční náklady	140
I.10	Grafana - Průběh teploty - Query A	142
I.11	Grafana - Průběh teploty - Query B	142
I.12	Grafana - Průběh teploty - Query C	142
I.13	Grafana - Průběh teploty - Query D	143

Úvod

S postupem času se rozšiřují možnosti využití elektrotechniky ve všech technických oblastech. Elektronika je dnes běžnou součástí většiny produktů, což se promítá i do moderních instalací. Oproti minulosti lze nyní ovládat nejen osvětlení, ale také topení, klimatizaci, žaluzie, zabezpečení budov, spotřebu energie a mnoho dalších funkcí. S rostoucí komplexností instalací dříve narůstalo i množství kabeláže, která sloužila převážně ke spínání jednotlivých spotřebičů. V současnosti lze tyto požadavky řešit pomocí napájecích zdrojů řízených sběrnicemi, což umožňuje automatizaci rozsáhlejších objektů, jako jsou hotely, nemocnice, vily nebo i továrny. Důležitým aspektem je také možnost vzdáleného řízení, které uživateli umožňuje ovládat celý systém odkudkoliv. Sběrnicové instalace se tak stávají stále populárnějším řešením, přičemž mezi nejrozšířenější standardy patří KNX, využívaný po celém světě.

Tato bakalářská práce si klade za cíl seznámit čtenáře se sběrnicovým systémem KNX, vybraným serverem pro řízení demonstračního panelu a tvorbou programu pro demonstraci funkcí tohoto systému. Panel bude využíván při různých akcích k prezentaci systému KNX, a to nejen prostřednictvím fyzických ovládacích prvků, ale také pomocí webového rozhraní programovatelného logického automatu nebo aplikace Home Assistant. Zároveň bude sloužit k prezentaci společnosti, které poskytly zařízení použité v panelu.

Teoretická část práce se zaměřuje na základy sběrnicového systému KNX, jeho historii, možnosti využití a principy fungování sběrnicových přístrojů včetně jejich adresace a komunikace. Dále je popsáno zabezpečení systému a topologie KNX. Následuje kapitola věnovaná použitému PLC, jeho knihovnám, komunikačnímu protokolu MQTT a vizualizačnímu serveru. Závěrem je popsána realizace vizualizace pomocí Raspberry Pi 5 s využitím kontejnerů (Portainer, Home Assistant, InfluxDB, Mosquitto, Grafana).

Praktická část práce se soustředí na tvorbu instalace v softwaru ETS, programování PLC a práci s Raspberry Pi. Nejprve je popsán software ETS a jeho možnosti, následně tvorba instalace od přidání zařízení do projektu, přes nastavení komunikace až po přiřazení skupinových adres. V části věnované PLC jsou popsány základní vytvořené funkční bloky, nastavení komunikace a realizace vizualizace prostřednictvím integrovaného webového serveru. Poslední kapitola se věnuje instalaci a nastavení Raspberry Pi, Dockeru a jednotlivých kontejnerů.

Cíle práce

Cílem této bakalářské práce je seznámení s technologií KNX, její nastavení pomocí softwaru ETS, vytvoření vzdáleného ovládání a vizualizace skrze programovatelný automat a rozebrání možností jiného přístupu k ovládání a vizualizaci skrze software s otevřeným zdrojovým kódem.

1 Sběrnicový systém KNX

Existuje velké množství sběrnicových systémů, ale asociace KNX s 500 členskými společnostmi a 8000 produkty je v současnosti největší na trhu. [10]

Pro vstup do asociace je nutné, aby bylo splněno několik kritérií [4]:

- **splňovat požadovanou kvalitu** – kompatibilita s ISO 9001
- **vzájemná kompatibilita s ostatními členy**
- **konfigurační kompatibilita** – možná konfigurace za použití KNX Engineering Tool Software, zkráceně ETS
- **zpětná kompatibilita** – kompatibilita starých instalací s nynějšími a budoucími instalacemi

Výhodou takto velké asociace je již zmíněná vzájemná kompatibilita komponent členských společností, tisíce KNX certifikovaných skupin výrobků (pokrytí jakéhokoliv myslitelného pole aplikací), podpora všech komunikačních médií (kroucený pár TP, powerline PL, radiofrekvenční RF a rozhraní IP/Ethernet/WLAN), použití jednoho softwaru (ETS) na projektování a programování všech výrobků členských společností. [4]

KNX je také normalizováno v Evropě, USA, Číně a mezinárodně prostřednictvím norem. Tyto normy zajišťují snadné rozšíření instalace, případně její výměnu za novou, nebo již zmíněnou kompatibilitu mezi společnostmi. [3]

1.1 Historie

1.1.1 EIBA

Asociace byla založena v Belgii roku 1990 pod názvem European Installation Bus Association (EIBA) se záměrem vytvářet instalace schopné komunikace pomocí sběrnic. Jako první komunikační médium byl použit TP. Pro zajištění kompatibility mezi produkty se členské společnosti dohodly na používání jednoho systému (standardu). Mezi další důležité milníky patří [2]:

- **1991** – první školení EIBA
- **1992** – první certifikované zařízení na trhu
- **1993** – představení první verze ETS na trhu
- **1994** – vznik prvních školících center
- **1996** – vznik The Scientific Partnership
 - PL použito jako komunikační médium

1.1.2 KNX

Roku 1999 se EIBA sloučila se společnostmi Batibus Club International (BCI) a European Home Systems Association (EHS). Přijaly název Konnex Association. Sídlem asociace byl ustanoven Brusel. Toto sloučení nemělo vliv na zpětnou kompatibilu, tudíž jsou všechny nové produkty kompatibilní s produkty nesoucími logo EIB. Důležité milníky pro KNX [2]:

- **2001** – vytvoření nového standardu KNX se základem ve standardu EIB
- **2003** – standard schválen jakožto evropská norma EN 50090
- **2004** – standard schválen, jako americká norma ANSI/ASHRAE 135
 - přidání přenosového média RF do standardu KNX
- **2006** – standard schválen, jako světová norma ISO/IEC 14543-3
 - přejmenování asociace na KNX
- **2007** – standard schválen, jako jedna z čínských norem GB/Z 20965
 - KNX IP bylo představeno jako čtvrté přenosové médium
- **2013** – standard schválen, jako jediná čínská norma GB/T 20965

1.2 Možnosti použití technologie

Použití inteligentní instalace umožňuje využití celého objektu s maximálním potenciálem a tím ulehčit uživateli práci. Níže jsou uvedeny příklady použití instalace KNX [7]:

- **Centrální ovládání** – možnost ovládat celou instalaci z jednoho zařízení (např. centrální panel, mobil)
- **Realizace centrálních funkcí** – při odchodu z domu zhasnutí světel, spuštění žaluzií, vypnutí zásuvkových obvodů....
- **Regulace teplot (topení, chlazení)** – regulace teploty každé místnosti zvlášť (lze nastavit vypnutí topení při otevření okna)
- **Režimy nastavených teplot** – tepelné režimy (Ekonomický, Komfort,...) chránící před přehřátím či promrznutím.
- **Světelné scény** – intenzita osvětlení, volba světel, případně i barva
- **Rozdělení místností na více obvodů**
- **Použití virtuálních asistentů** – ovládání instalace hlasovými povely přes virtuální asistenty (Alexa, Google Home,...)
- **Simulace přítomnosti** – simulace přítomnosti v objektu pomocí ovládání žaluzií a osvětlení
- **Kontrola spotřeby energií** – monitorování a optimalizace spotřeby energií

1.3 Sběrnicové instalace

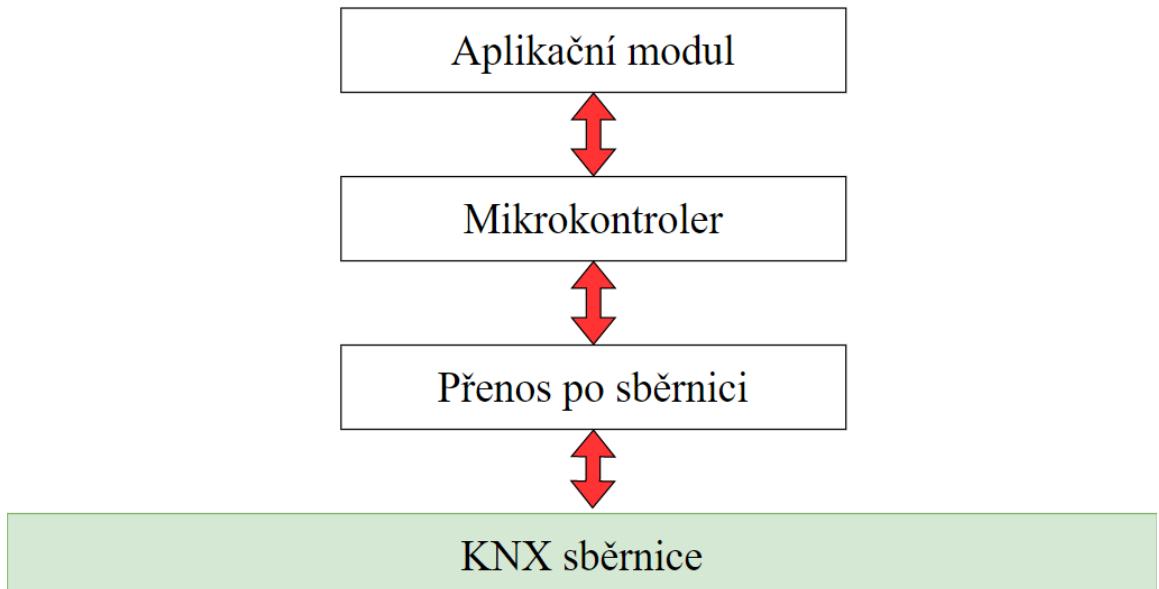
Sběrnicová instalace je založená na koncepci ICT (Information and Communication Technology). Třemi hlavními aspektami této koncepce jsou [4]:

- náhrada klasických spínačů tlačítkovými ovladači schopnými komunikace, nebo připojení klasických spínačů k rozhraním schopných komunikace
- připojení rozhraní se schopností komunikace, nebo nepřímého ovládání (spínačí přístroje schopné komunikace) ke všem spotřebičům
- propojení veškerých přístrojů schopných komunikace kabelem určeným na bezpečné malé napětí

1.3.1 Sběrnicové přístroje

Zařízení připojené ke sběrnici se schopností komunikovat s dalšími přístroji se nazývá sběrnicovým přístrojem a je tvořeno těmito částmi (viz Obr. 1.1) [4]:

- **Přenosový modul** – vytváří rozhraní pro přenos informací
- **Mikrokontroler** – komunikace mezi přenosovým modulem a aplikačním modulem
- **Aplikační modul** – obvod tvořící přístroj¹



Obr. 1.1: Součásti sběrnicového přístroje [4]

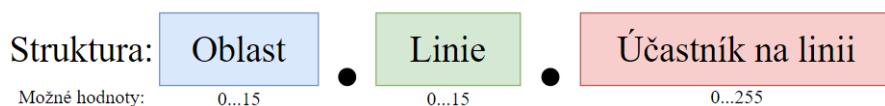
¹Spojení přenosového modulu a mikrokontroleru tvorí tzv. sběrnicovou spojku (bus coupler unit BCU).

Přístroje lze ještě dělit na aktivní a pasivní. Pasivní přístroje nejsou součástí ICT, ale jedná se o podpůrné přístroje určené pro podporu procesu (nekomunikují s ostatními přístroji). Jedním z příkladů pasivních přístrojů jsou napájecí zdroje. Příkladem pasivních přístrojů jsou napájecí zdroje. Napájecí zdroje mohou být rozšířené ještě o ICT, ale není to časté. Aktivní přístroje lze rozdělit do těchto kategorií [4]:

- **Rozhraní** – propojuje sběrniči a PC
- **Spojky** – optimalizují komunikaci v systému
- **Snímače** – předávají informace sběrnicovému systému
- **Akční členy** – propojují klasické spotřebiče se sběrnicovým systémem

1.3.2 Adresování

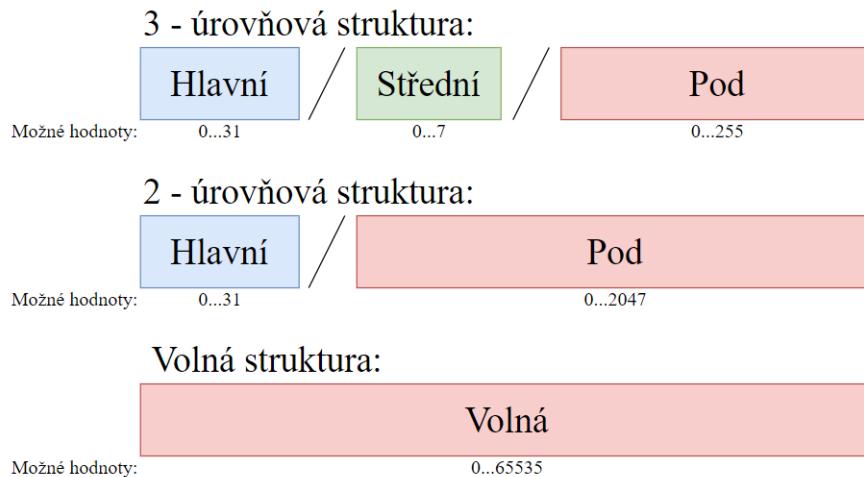
Individuální adresa je v instalaci jedinečná, tj. neexistuje další stejná adresa a požívá se k přesné identifikaci přístroje na sběrniči. Adresa je 16-bitová a je rozdělená na tři části (viz Obr. 1.2).



Obr. 1.2: Struktura individuální adresy [6]

Nastavování individuální adresy na přístroji probíhá většinou stiskem programovacího tlačítka na přístroji. Při stisknutí tlačítka se rozsvítí programovací LED. Individuální adresa se přístroji přiděluje natrvalo. Po přidělení již ETS posílá příslušná data (aplikace, konfigurace, parametry, skupinové adresy).

Při uvedení do provozu probíhá komunikace pomocí skupinových adres. Jedná se o adresy definované programátorem pro každou funkci v systému. Celkově je možno použít 65535 adres, přičemž adresa 0/0/0 je rezervována pro tzv. broadcast (hlášení všem přístrojům na sběrniči). Programátor si také může zvolit, kterou z uvedených struktur použije (viz Obr. 1.3). Nejčastěji se využívá třístupňová struktura kvůli přehlednosti. Hlavní skupina se používá na číslo podlaží, střední skupina na funkci (např. 1 = osvětlení, 2 = topení, 3 = stínění etc.) a podskupina pro konkrétní spotřebič nebo skupinu spotřebičů. [6]

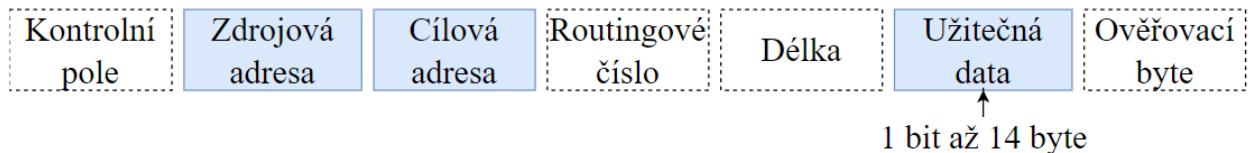


Obr. 1.3: Příklad struktury skupinových adres [6]

1.3.3 Komunikace po sběrnici

Komunikace přístrojů na sběrnici probíhá pomocí tzv. telegramů (viz Obr. 1.4), kde je délka dat závislá na typu datového bodu (1bit - 14bytů). Nejdůležitějšími částmi telegramu jsou tři bloky [6]:

- **Zdrojová adresa** – udává adresu přístroje, který telegram vyslal
- **Cílová adresa** – adresa přístroje, kterému je telegram určen
- **Užitečná data** – příkaz, který má daný přístroj vykonat

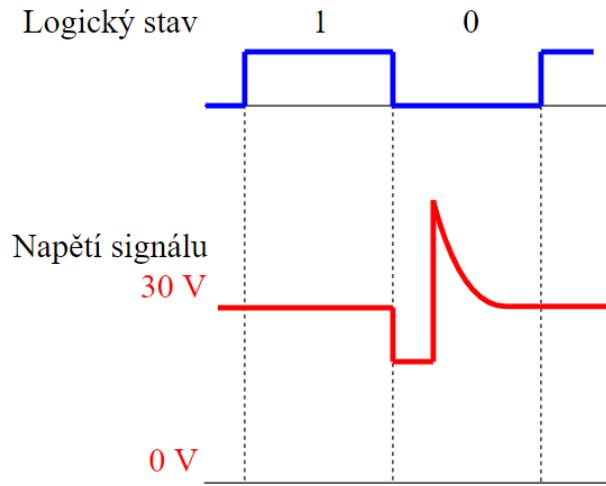


Obr. 1.4: Struktura telegramu [6]

Telegramy na sběrnici čtou všechny přístroje, ale vykoná jej pouze přístroj určený cílovou adresou.

Komunikace na sběrnici probíhá pouze v případě, že je na sběrnici logická "1". V opačném případě je sběrnice přeplněná a pokračuje ve vysílání pouze přístroj s logickou "0" (viz Obr. 1.5). [6]

Aby se zabránilo kolizím a umožnilo vysílání pouze jednomu přístroji, je přenos řízen metodou CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance — vícenásobný přístup s detekcí nosné a vyhýbáním se kolizím). Přístroj, který chce



Obr. 1.5: Struktura bitu kroucené dvojlinky [6]

vysílat logickou „1“, nejprve sleduje sběrnici; pokud detekuje logickou „0“, přenos přeruší a vyčká na uvolnění sběrnice. Po ukončení přenosu jiného zařízení se pokusí vysílat znovu. [6]

1.3.4 KNXnet/IP

Pro překlenutí fyzické sběrnice KNX a IP sítí slouží protokol *KNXnet/IP*, který definuje dva základní režimy:

Tunelování (Tunnelling)

Tunelování umožňuje bod–bod komunikaci mezi KNX sběrnicí a IP klientem. KNX telegramy se zabalí do UDP/IP paketů a posílají na port 3671. Server udržuje pro každé TCP/UDP spojení vlastní *Individual Address*, kterou klientovi přidělí při navázání tunelu. Z toho v praxi vyplývají tyto tři možné režimy [12]:

- **Sdílená adresa (Shared IA)**

Režim A používá KNXnet/IP Server **stejnou Individual Address** pro všechny tunelovací relace. Klienti tedy sdílí mezi sebou jeden fyzický adresní bod na TP-lince. Výsledkem je jednoduchá konfigurace, ale horší rozlišovací schopnost mezi klienty posílajícími telegram.

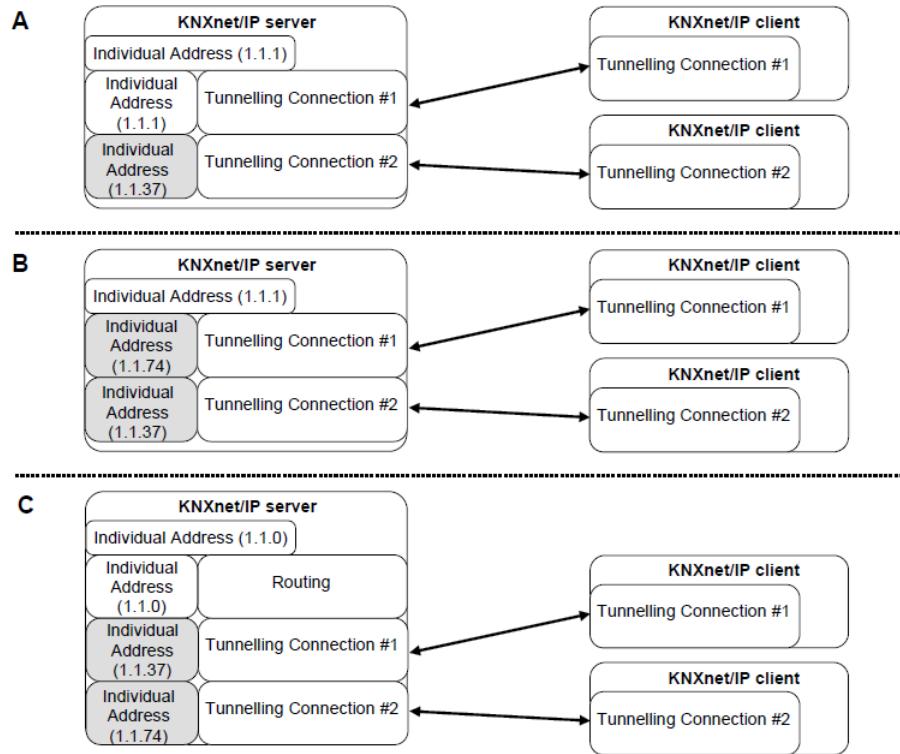
- **Dedikovaná adresa (Dedicated IA)**

Režim B přidělí každému tunelovacímu klientovi **unikátní Individual Address**. Tím lze monitorovat a spravovat každou relaci individuálně.

- **Kombinovaný režim (Routing + Tunnelling)**

V Režimu C server kromě přidělení dedikovaných tunelů funguje i jako KNX/IP

router pro multicast skupinových telegramů. Má tedy jednu IA pro routování (např. 1.1.0) a další IA pro tunnelling relace. Díky tomu zvládne jak point-to-point servis a diagnostiku, tak i point-to-multipoint přenosy mezi subnety.



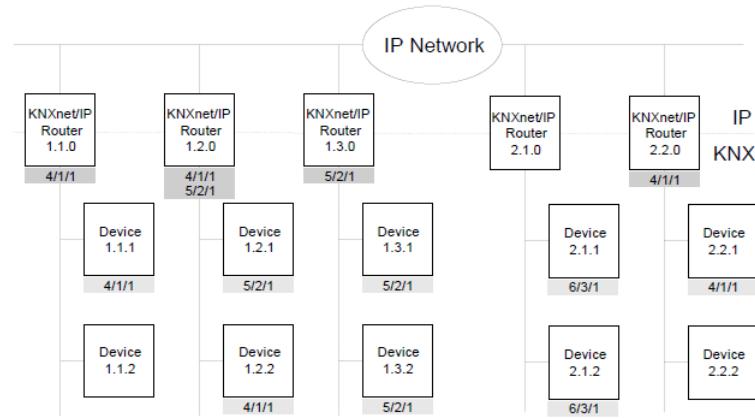
Obr. 1.6: Tunelování [12]

Routování (Routing)

Routing zajišťuje přenos skupinových (point-to-multipoint) telegramů mezi více KNX líniovými segmenty (linky/subnety). Každý KNXnet/IP Router přijímá příchozí telegramy na multicastové skupině a podle filtrovaných adres je znova vysílá na KNX sběrnici. Díky tomu umožňuje velmi rychlé doručení telegramů současně do více destinací. [13]

1.3.5 Datový bod

Datové typy byly standardizovány za účelem zajištění kompatibility podobných přístrojů od různých výrobců. Jedná se například o stmívání, žaluzie a hodiny. Standardizace zahrnuje požadavky na formát dat a strukturu komunikačních objektů snímačů a akčních členů. I tak existuje více druhů datových bodů (DPT) se stejnou



Obr. 1.7: Routování [13]

funkcionalitou. Kombinace různých typů DPT se nazývají funkčními bloky. [6]

Skutečná informace datového bodu:

- Není uložena v paměti zařízení.
- Není nikdy součástí telegramu
- Je pouze v projektu ETS

Typy datových bodů jsou zvláště důležité pro diagnostiku. To znamená, že umožňují ETS monitorovat data spojená se skupinovými objekty, např. místo "data = 85 A8" je zobrazeno "data = -6 °C". [1]

Struktura datového bodu je tvořena dvěma částmi a zapisuje ve tvaru X.YYY, nebo DATOVÝ TYP.VELIKOST. Tyto části jsou [1]:

- **Datový typ** – formát + kódování
- **Velikost** – rozsah hodnot + jednotky

Tab. 1.1: Přehled nejpoužívanějších datových typů

Značení	Formát	Funkce
1.yyy	boolean	přepínání (001), krok (007),...
3.yyy	boolean + 3-bit unsigned	stmívání
5.yyy	8-bit unsigned + 3-bit unsigned	stmívání(0-100%), pozice rolet(0-100%)
7.yyy	boolean + 3-bit unsigned	čítač pulsů
9.yyy	16-bit float	přenos hodnoty teploty, jasu, rychlost větru
14.yyy	32-bit float	nastavení teploty
19.yyy	čas + data	výstupy obrazovek
20.yyy	8-bit enumerace	Topení, chlazení a ventilace ('komfort',...)

Důsledkem existence datového bodu je schopnost nastavit hodnotu osvětlení 3 různými způsoby [7]:

- **Zapnutí/Vypnutí**
- **Krokové stmívání** – Při poslání telegramu "start stmívání" osvětlení krokově roste o definovanou hodnotu. Po poslání "stop stmívání" hodnota neroste.
- **Procentuální stmívání** – Realizuje se pomocí cyklického posílání telegramu. Při každém přijetí telegramu se zvedne jas o nastavenou hodnotu.

1.4 Zabezpečení

Rozdíl mezi zařízeními KNX a zabezpečenými KNX Secure je ten, že zařízení KNX Secure jsou schopna šifrovat a dešifrovat telegramy. Tato technologie dodává instalaci extra zabezpečení, a to během uvádění instalace do provozu, tak i poté za běhu. Telegramy jsou zašifrované zabezpečenými zařízeními KNX a nazývají se zabezpečené telegramy.

Lze rozlišit dva typy šifrovaných telegramů KNX [5]:

- **Zcela zašifrované**
 - lze použít pouze na zařízeních KNX/IP (KNX IP Secure)
 - používá se, pro zabezpečení části instalace, která je vystavená externí IP síti (typicky se jedná o páteřní linku)
- **Částečně zašifrované**
 - lze požít na libovolné komunikační zařízení KNX (KNX Data Secure)
 - používá se pouze pro instalaci, která není vystavena exteriéru IP sítě

Oba typy zabezpečení obsahují MAC (Message Authentication Code).

Zabezpečená zařízení mají zabezpečený režim, který je v projektu ETS reprezentován vlastností nazvanou „Secure Commissioning“. Pouze když je tento režim aktivován, zařízení je schopno šifrovat a dešifrovat telegramy.

Zabezpečená zařízení mají tzv. "Tool Key". V momentě, kdy je aktivován zabezpečený režim zařízení, je ETS schopen komunikovat s tímto zařízením pouze pokud zná Tool Key tohoto zařízení.

Zabezpečená zařízení obsahují také Factory Default Setup Key (FDSK). FDSK je jedinečný pro každé zařízení a nelze jej upravovat ani mazat. ETS tento klíč může načíst jenom pomocí certifikátu (25znakový kód, který obsahuje sériové číslo a FDSK). Tool Key je z výroby nastaven na FDSK. Tool Key může být také zpětně nastaven na FDSK pomocí tzv. "master resetu", který uvadí výrobce.

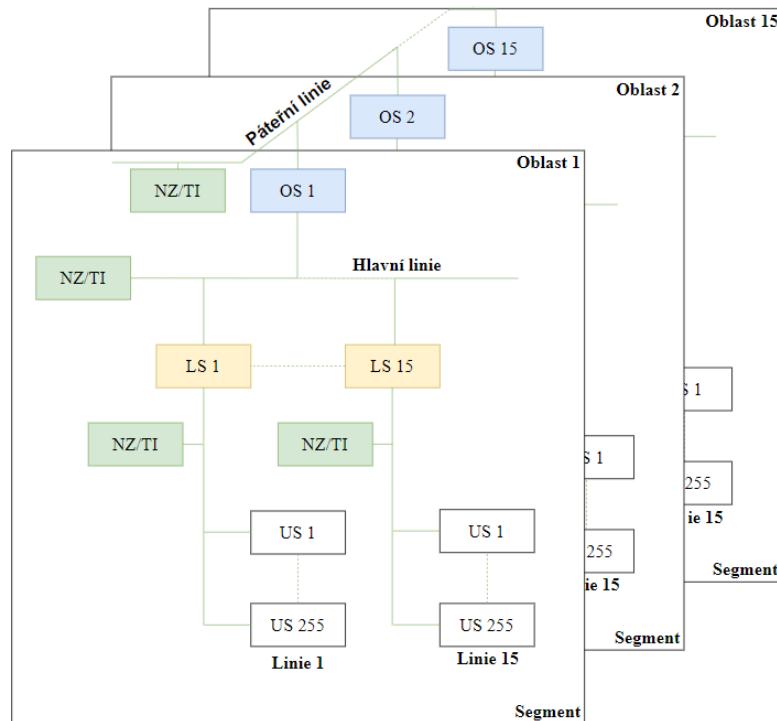
Po přidání zabezpečeného zařízení KNX do ETS a po přidání jeho certifikátu, ETS automaticky nastaví svůj Tool Key v projektu. To znamená, že uživatel ETS nemůže definovat/upravit Tool Key ručně, Tool Key také není viditelný pro uživatele ETS. [5]

1.5 Topologie

Základním kamenem topologie je hlavní linie, na kterou lze připojit až 256 přístrojů (účastníků sběrnice - US). Tato linie lze rozdělit až na 15 dalších segmentů za použití líniových opakovačů/spojek (LS). Na takto vzniklé segmenty (linie) připojit dalších 256 US. To vše ovšem závisí také na spotřebě přístrojů použitých v instalaci. To znamená, že celková spotřeba všech přístrojů nesmí překročit jmenovitý proud na druhé straně sběrnicového zdroje, který každá linie musí mít vlastní. Také lze mít maximálně 4 000 US na celé topologii. Toto množství lze také navýšit za použití oblastní spojky (OS) díky na páteřní linii. Po připojení vznikne tzv. nadřazená páteřní linie, která může pojmut až 16 oblastních spojek a celek rozdělí na dílčí páteřní linie. Celkový počet US na takovéto linii může být až 61 000. Reálné množství je v tomto případě omezeno zdrojem s tlumivkou (NZ/TI). [8]

Pro sběrniči KNX lze použít pouze tyto struktury kabeláže:

- Hvězdicová
- Líniová
- Stromová
- Kombinace výše uvedených



Obr. 1.8: Ukázka topologie KNX[8]

1.5.1 Individuální adresa

Individuální adresa se nastavuje s ohledem na umístění v topologii (viz Podkapitola 1.3.2).

Tab. 1.2: Individuální adresy v topologii [8]

Prvek	Adresa	Funkce
Oblast	0	adresuje účastníky v páteřní linii
Oblast	1...15	adresuje oblasti
Linie	0	adresuje hlavní linii příslušné oblasti
Linie	1...15	adresuje linie obsažené v oblasti
Účastník na sběrnici	0	adresuje liniovou spojku příslušné linie
Účastník na sběrnici	1...255	adresuje sběrnicové přístroje obsažené v linii

1.5.2 Spojka

V případě, že jsou v instalaci použity spojky a mají přiřazeny správné individuální adresy, budou při projektování v programu ETS (Kapitola 2) automaticky vytvořeny filtrační tabulky jednotlivých spojek. Filtrační tabulka obsahuje skupinové adresy, které smí projít skrz příslušnou spojku (obsahuje všechny obsažené skupinové adresy, které adresují SU umístěné za spojkou). Tudíž každá linie pracuje nezávisle.

Spojky jsou vytvořeny pro montáž na DIN lištu, kde se připojují primární i sekundární linie pomocí sběrnicové svorkovnice. Primární linie také funguje, jako napájení mikrokontroleru a při výpadku sítě ohlásí tuto skutečnost na sekundární linii. Jednou z výhod spojky je možnost programování z obou linií. Obsahují také žluté signalizující LED, které blikají pouze v případě, že spojka propustí telegram na příslušnou linii. Další vlastností spojky je galvanické oddělení mezi primární a sekundární linií. Poslední vlastností spojky je možnost přeměny na liniový opakovač. Opakovač se rozliší od spojky absencí nuly na konci individuální adresy (X.X.1 apod.). Využívá se pro rozšíření linie o další segment s 64 US. Tento úsek je limitován délkou kabelu, který může měřit maximálně 1000 metrů. [8]

1.5.3 Routingové číslo

Každý telegram, který je vyslán přístrojem, obsahuje routingové číslo, které začíná na hodnotě 6. Toto číslo se při každém průchodu spojkou či opakovačem dekrementuje, dokud nedosáhne nulové hodnoty. Tuto vlastnost berou filtrační tabulky v potaz. Pokud se jedná o servisní telegram, tak routingové číslo má hodnotu 7, která se

při průchodu spojkou nedekrementuje.² Tuto skutečnost berou v potaz i filtrační tabulky, které toto číslo ignorují, a tudíž všechny spojky tento telegram propustí. Tento telegram se vždy dostane k požadovanému účastníku bez ohledu na umístění. Toto číslo také brání zasmyčkování (nekonečnému kolování) telegramu. [8]

1.5.4 Interní a externí rozhraní

Systém KNX je otevřený jiným systémům za použití vhodných rozhraní umístěných na libovolné linii (většinou se jedná o páteřní linii). Lze připojit například programovatelný logický automat (PLC), digitální síť integrovaných služeb (ISDN), systémová technika budov, internet a mnohé další. Tato rozhraní přenáší obousměrně zprávy, které převede na komunikační protokol.

Nejedná se ovšem jenom o spojovaní KNX s externími médii, ale je možno spojit různá KNX média mezi sebou (např. spojení TP, RF, optika) [8].

²Spojky vyrobeny po roce 2019 mají schopnost tuto hodnotu dekrementovat

2 ETS

Jedná se o konfigurační softwarový nástroj, který je nezávislý na výrobci pro navrhování, konfiguraci inteligentních instalací, a také pro řízení budov pomocí systému KNX. Tento software funguje pouze na počítačových platformách využívajících operační systém Windows. [11].

Pomocí softwaru lze [9]:

- **Vkládat katalogové produkty do projektu** – Produkty schválené asociací KNX jsou obsaženy v katalogu a lze je použít v projektu. Produkty lze také přidávat manuálně prostřednictvím aplikačních programů s koncovkou ".knxprod".
- **Vytvořit architekturu objektu** – rozdělit objekt na celky(budovy, patra, místnosti,...)
- **Parametrizace produktů**
- **Vytváření skupinových adres**
- **Nahrávání řešení projektu do přístrojů**
- **Vzdálené ovládání připojeného projektu**
- **Diagnostika**
- **Vytvoření dokumentace**

2.1 Tvorba instalace

Při vytváření projektu byl zvolen typ páteřní linie na IP, skupinové adresy byly zvoleny třístupňové a topologie typu TP. TP topologie byla použita i při tvorbě panelu.

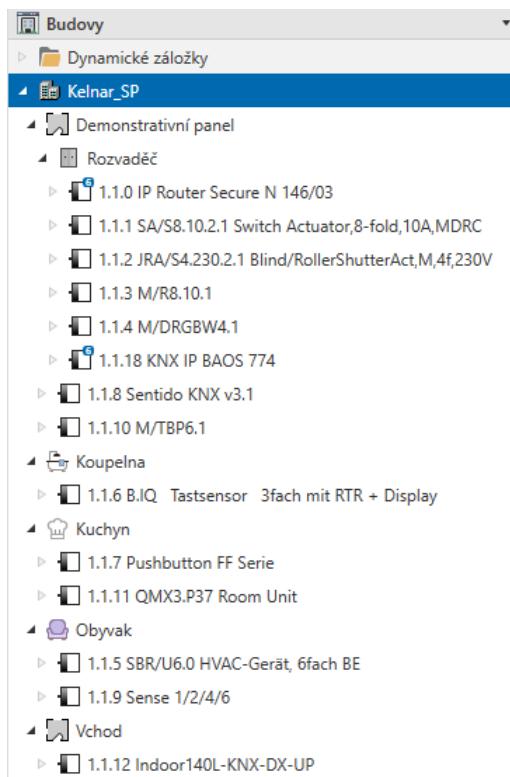
Po úspěšném založení projektu se program přepnul do pracovní části, která je složená z osmi oken [9]:

- **Budovy** - rozdělení objektu na celky
- **Skupinové adresy** - vytvoření a přiřazení skupinových adres přístrojům
- **Topologie** - zobrazení rozložení vytvořeného projektu v topologii
- **Kořeny projektu** - zobrazení všech oken kde se pracovalo
- **Přístroje** - seznam přístojů v projektu
- **Zprávy** - okno zaměřené na tvorbu dokumentace projektu
- **Katalog** - vyhledání a vložení produktů do projektu
- **Diagnostika** - okno určené pro otestování instalace

Pro vytvoření pracovního prostoru bylo použito okno *budova*. Prostor byl pojmenován *Demonstrativní panel* a byl rozdělen na pět logických celků. Tyto celky reprezentují pokoje zobrazené na panelu (vchod, kuchyň, koupelna, obývací pokoj a

rozvaděč umístěný v zadní části panelu). Toto rozdelení bylo vytvořeno za účelem zvýšení přehlednosti objektu a následné ulehčení propojování přístrojů mezi sebou. Je nutno také dodat, že vytvoření jedné místnosti je podmínkou pro vkládání přístrojů do pracovní plochy.

Pro vložení přístrojů bylo nutno otevřít okno katalog, který ovšem neobsahoval použité přístroje. Kvůli této komplikaci bylo nutno navštívit webové stránky výrobců a následné stažení aplikačních programů. Tyto programy byly importovány do katalogu pomocí tlačítka *Import....*. Vzhled projektu po přidání přístrojů lze vidět na Obr. 2.1.



Obr. 2.1: Projekt budovy v ETS

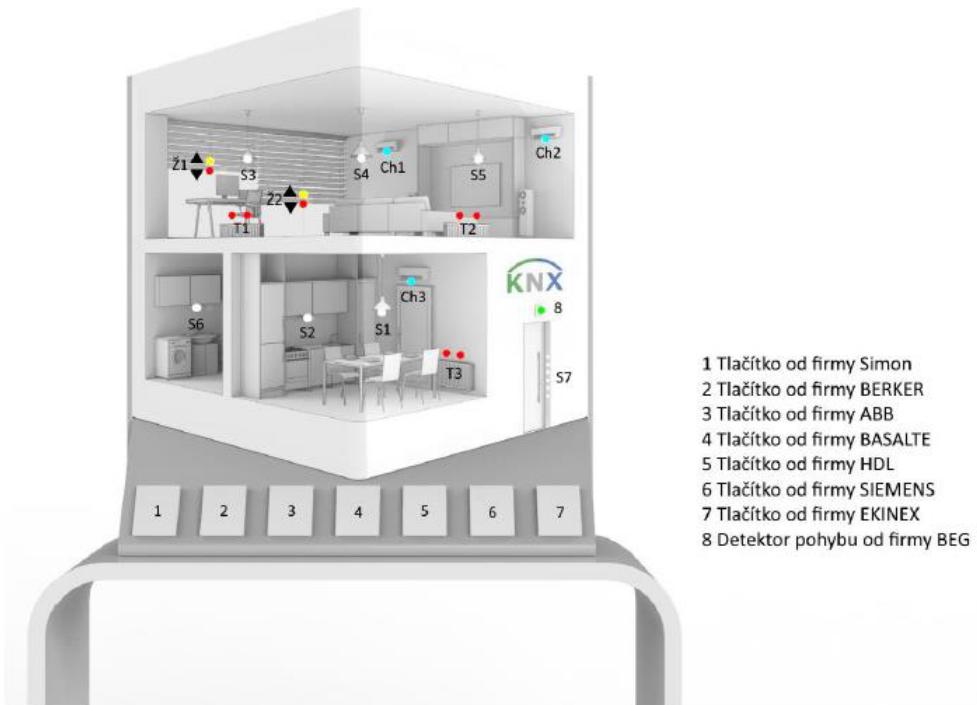
Po přidání všech přístrojů se zobrazila pracovní plocha, která slouží k zobrazení přehledu všech přístrojů (Zabezpečení - KNX Secure, individuální adresa prvku, místnost v projektu, použitý aplikační program, stav přístroje - nahrána adresa, program, parametrizace, skupinová adresa a informace o produktu). Ve sloupcích vyjadřujících stav přístroje jsou většinově pomlčky, které znázorňují, že nebyly nahrány všechny části do přístrojů. Tahle skutečnost je zdůvodněná změnami parameterů a skupinových adres.

Zábez	Adresa	Místnost	Popis	Aplikační program	Adr	Prg	Par	Grp	Cfg	Výrobce	Objednací	Produkt
1.10	Rozvaděč			091A.CO IP Router Secure 004003	-	-	-	-	-	Siemens	5WG1146...IP Router Secure N 146/03	
1.141	Rozvaděč			Switch 8f 10A/3.2b	✓	-	-	-	✓	ABB	2CDG 110 1...SA/SB.10.2.1 Switch Actuator;8-fold,10A,MDR	
1.142	Rozvaděč			Blind/Roller Shutter 4f 230V M/1.4	✓	-	-	-	✓	ABB	2CDG 110 1...JRA/S4.230.2.1 Blind/RollerShutterAct,M,4f,23...	
1.143	Rozvaděč			Switch 8fold 10A (V1.2)	✓	-	-	-	✓	HDL	M/RB 1105...M/RB.10.1	
1.144	Rozvaděč			RGBW 4fold Driver(V1.0)	✓	-	-	-	✓	HDL	M/DRGBW...M/DRGBW4.1	
1.145	Obyvák			HVAC device, 6gang BE/1	✓	✓	✓	✓	✓	ABB	SBR/U6.0 SBR/U6.0 HVAC-Gerät, 6fach BE	
1.146	Koupelna			B/IQ Multifunkční RTR + Display V2.161302	-	-	-	-	✓	Berker	7566359x B/IQ Tastsensor 3fach mit RTR + Display	
1.147	Kuchyn			APEKED2TP	✓	✓	✓	✓	✓	Ekinex S.p.A.	EK-ED2-TP Pushbutton FF Serie	
1.148	Demonstrativní panel			Sentido KNX app v3.1	✓	✓	✓	✓	✓	BASALTE	200-02 Sentido KNX v3.1	
1.149	Obyvák			Dsense	✓	✓	✓	✓	✓	Simon	8400100-0... Sense 1/2/4/6	
1.150	Demonstrativní panel			Touch 6buttons panel(1.1)	✓	✓	✓	✓	✓	HDL	M/TBP6 13...M/TBP6.1	
1.151	Kuchyn			QMX3.P37	✓	✓	✓	✓	✓	Siemens HVAC	QMX3.P37 QMX3.P37 Room Unit	
1.152	Vchod			DX_V6.x	✓	✓	✓	✓	✓	B.E.G.	93393 Indoor140L-KNX-DX-UP	

Obr. 2.2: Pracovní plocha v ETS

2.2 Parametrizace tlačítek a detektoru pohybu

V této podkapitole je vysvětleno parametrizování použitých tlačítek. Ta byla pomyslně rozdělena do místností a nastavena, tak aby spolupracovala s nejbližšími prvky (světly, žaluziemi, klimatizací a topením), které jsou zobrazeny na Obr. 2.3. Pro vysvětlení byly vytvořeny tabulky popisu funkcí jednotlivých tlačítek.



Obr. 2.3: Grafický návrh panelu [9]

2.2.1 ABB - SBR/U6.0.1-84

Jedná se o šestinásobné tlačítko se zabudovaným termostatem, které lze použít na regulaci teploty, ovládání žaluzií, ovládání osvětlení a nastavení dvou scén, které mohou obsahovat až osm objektů. Každé stisknutí tlačítka změní barvu signalizační LED na předem stanovenou hodnotu (rozpoznání zapnuto/vypnuto). [14]



Obr. 2.4: Šestinásobné tlačítko s termostatem ABB - SBR/U6.0.1-84 [14]

Tlačítko bylo nastaveno na odesílání aktuální hodnoty teploty každých deset minut. Tlačítka jsou rozložena po horizontálních párech s označením funkční blok 1 až 3. V záložce každého bloku byla nastavena obě tlačítka na krátká a dlouhá stisknutí. V záložkách *Common parameter* byl vybrán typ objektu na 1-bit. Při krátkém stisknutí tlačítka odesílá hodnotu 1, při dlouhém stisknutí posílá hodnotu 2. Následně v záložce *Extended parameters* byly nastaveny hodnoty odesílaných objektů u dlouhého stisknutí na on ("1") a u krátkého na off ("0"). V záložkách *LED Button* pro každý funkční blok byla každá dioda nastavena do modu status. Přijímaný objekt byl nastaven na 1-bit a hodnota jasu na *bright* signalizační barvu LED diod, tedy bílou při vypnutí a červenou při zapnutí.

2.2.2 Berker - 75663593

Osminásobné tlačítko s termostatem by mělo být schopno regulovat pokojovou temperaturu, ovládat žaluzie, ovládat osvětlení a scény. V této práci se ovšem nepovedlo nastavit ani při použití více zařízení a softwaru od Berkeru, který dokázal otevřít externí okno parametrizace v německém jazyce. Po ukončení parametrizace se parametry neuloží. [19]



Obr. 2.5: Osminásobné tlačítko - Berker - 75663593 [19]

2.2.3 Ekinex - EK-ED2-TP-RW

Jedná se o čtyřnásobné tlačítko se zabudovaným teplotním senzorem pro ovládání žaluzií, osvětlení a scén. [20]

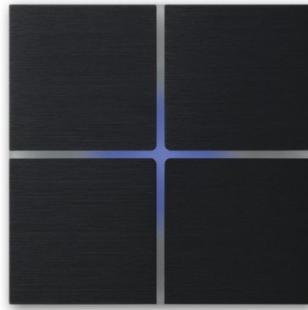


Obr. 2.6: Čtyřnásobné tlačítko Ekinex - EK-ED2-TP-RW [9]

Tlačítko bylo nastaveno v záložce *General* na dvě svislé klapky. Obě klapky byly nastaveny na dlouhé a krátké stisknutí. V případě první klapky se horní krátký stisk nastavil na funkci *toggle* (přepínání). Dlouhý stisk představuje funkci *off*. Pro dolní část klapky, je nastavení přesně opačné. Druhá klapka je nastavená stejným způsobem, akorát místo funkce *toggle* byla použita funkce *none*. Tato funkce zasílá "0", což znamená u žaluzií pohyb směrem nahoru.

2.2.4 Basalte - Senido 202-03

Další z použitých snímačů je čtyřnásobné dotykové tlačítko se zabudovaným snímačem teploty pro ovládání žaluzií, osvětlení a scén. Toto tlačítko má rovněž schopnost rozlišovat krátké a dlouhé stisknutí, a to nejen na jednom segmentu, ale má možnost snímat více segmentů najednou (multitouch). Dokáže ovládat až šest scén s osmi objekty. Další ze schopností tlačítka je posílání tříbajtové hodnoty RGB. Poslední z funkcí tlačítka je zobrazování statusu díky RGB podsvícení. [17]



Obr. 2.7: Čtyřnásobné dotykové tlačítko Basalte - Senido 202-03 [17]

Tlačítko bylo rozděleno na čtyři samostatné zóny v záložce *General*. Dále se v této záložce povolila funkce řadiče scén. První trojici tlačítek byla nastavena scéna, kterou budou při stisknutí volat. Každá z těchto scén byla nastavena v korespondující záložce označené číslem. Poslední tlačítko bylo nastaveno na demonstraci schopnosti zasílat hodnoty RGB. Jedná se o 2 nastavené hodnoty, které se rozlišují délkou stisku. Pro demonstraci funkce multitouch byla vybrána funkce *room toggle + General on/off/scene*. Pro krátký stisk byla vybrána scéna, která se zapne při krátkém stisku. Při druhém stisku se panel vypne. Dlouhý stisk má přiřazenou vlastní scénu. V záložce *Temperature senzor* bylo nastaveno, aby senzor zasílal teplotu každých 5 minut. Záložka *Scene controller*, která je určena pro nastavení řadiče scén, byla nastavena na všech výstupech na hodnotu 1-bit.

2.2.5 Simon - 8400100-039

Čtyřnásobné tlačítko se zabudovaným RGB podsvícením a teplotním senzorem pro ovládání žaluzií a osvětlení. [26]

V záložce *General* bylo vybráno čtyř tlačítkové provedení, které je použito na demonstrativním panelu. Jako další možnost, která byla povolena, byl vnitřní senzor teploty. Poté v záložce *FeedBack* byly nastaveny hodnoty jasu a hlasitosti na maximum. Dále zde byla aktivována možnost zpětné vazby doteku vibracemi. Pro nastavení samotné funkcionality tlačítek se musela použít záložka *Inputs*, kde se

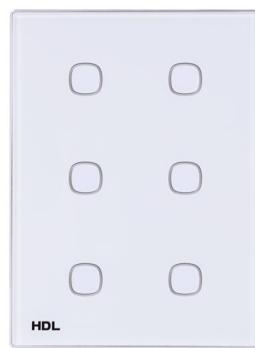


Obr. 2.8: Čtyřnásobné tlačítko Simon - 8400100-039 [26]

nastavilo oddělení tlačítek od sebe (všechna tlačítka jsou samostatně). Tlačítka v tomto případě jsou číslována od spodního levého rohu po sloupcích (1 a 2 levá strana, 3 a 4 pravá strana). Poté už se nastavovala samotná funkcionalita tlačítek. Byla vybrána možnost krátkého i dlouhého stisku. V případě krátkého stisku žaluzie vyjedou/sjedou samostatně. Dlouhý stisk znamená pohyb pouze v čase, kdy je tlačítko stisknuto.

2.2.6 HDL - M/TBP6.1-A2

Předposlední tlačítko je od společnosti HDL. Jedná se o šestinásobné dotykové tlačítko se zabudovaným RGB podsvícením pro ovládání žaluzií, osvětlení, stmívání a ovládání dvou scén s deseti objekty. Dále také obsahuje RGB kontrolér, který dokáže posílat 3-byte hodnotu obsahující informace o intenzitě každé složky. [21]

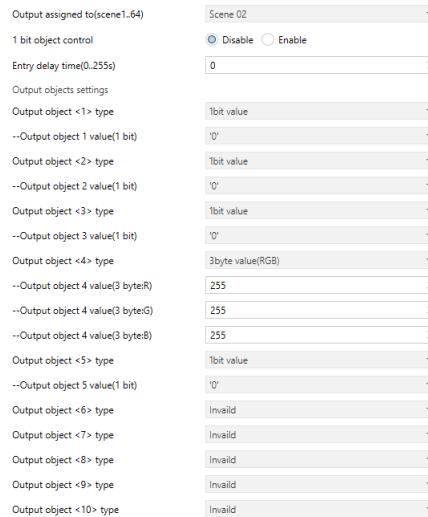


Obr. 2.9: Šestinásobné tlačítko HDL - M/TBP6.1-A2 [21]

První z parametrů, které je možno nastavit v záložce *General* byla citlivost dotyku, a to na hodnotu 4. Dále se pak povolily scény. Následně se obě scény nastaví v záložkách *Panel scene A* a *Panel scene B*. První scéna byla nastavena dle Obr. 2.10 a druhá dle Obr. 2.11.



Obr. 2.10: Parametry scény A tlačítka HDL - M/TBP6.1-A2



Obr. 2.11: Parametry scény B tlačítka HDL - M/TBP6.1-A2

Každé z těchto tlačítek má nastaveno signaliční podsvícení na jinou hodnotu. Krátkému stisknutí byla přiřazena funkce *toggle*, která dovoluje přepínat osvětlení mezi hodnotami zapnuto a vypnuto. Dlouhé stisknutí bylo nastaveno na dobu 1s a používá se na stmívání. Pro demonstraci stmívání byly nastaveny různé hodnoty

kroku prvních 4 tlačítek. Každé z těchto tlačítek má nastavenou signaliční podsvícení na jinou hodnotu. První tlačítko bylo nastaveno červenou barvu, druhou na zelenou, třetí na modrou a čtvrté na bílou. Při signalizaci se zvýší jas barev o 70%. Zbylá 2 tlačítka byla přepnuta do modu RGB kontrolér, který odesílají hodnotu RGB jak pro krátké, tak i pro dlouhé stisknutí. Tato hodnota se také signalizuje při stisknutí tlačítka.

2.2.7 Siemens - QMX3.P37

Jedná se o ovládací panel určený na regulování pokojové teploty s integrovaným displejem. Tento displej dokáže zobrazovat vlhkost vzduchu, koncentraci CO₂ v ovzduší a samotnou teplotu místnosti. Také obsahuje osm tlačítek, která obsahují žluté statusové LED. Tento panel umožňuje také ovládání žaluzií, osvětlení a scény. [24]



Obr. 2.12: Ovladací panel Siemens - QMX3.P37 [24]

V tomto případě bylo zařízení nastaveno na spínání pomocí jednoho tlačítka. Nejprve v záložce *General* byla nastavena hodnota svitu signalačních LED na 100% hodnotu. V další záložce byl nastaven teplotní senzor na odesílání hodnoty každých 10 minut. Poté se už nastavovaly jednotlivé tlačítkové páry. Funkce páru byla zvolena *Individual*, která umožňuje nezávislé fungování obou tlačítek. Dále se u obou tlačítek nastavila možnost *1 - button switching / send value, Short/long press* (dlouhé stisknutí po uplynutí půl sekundy) a vybrala se možnost odesílání druhé hodnoty při dlouhém stisku. Levým tlačítkem byla přiřazena hodnota *on* a pravým *off*. Také byla nastavena signalizace stisku tlačítek. Kvůli tomu byla možnost *LED display* nastavena na *status object* a možnost *LED activation* pro levá tlačítka na *0 = LED off; 1 = LED on*. Pravá tlačítka byla nastavena *0 = LED on; 1 = LED off*.

Po pozdější úvaze o zefektivnění panelu se pro 1. a 4. řadu tlačítek změnil dlouhý stisk na *Toggle*.

2.2.8 B.E.G - Indor 140-L-KNX-DX

V první záložce *Grundeinstellungen* (Základní nastavení) se nastavila hodnota teploměru (Temperaturmessung) na aktiviert. [18]

Parametrizace tohoto prvku byla celá v němčině, a to dosti zkomplikovalo postup. První záložka *Grundeinstellungen* (Základní nastavení) se nastavila hodnota teploměru (Temperaturmessung) na aktiviert. Poté v záložce teploměru se v možnosti zaslání teploty (*Temperaturwertsenden*) zvolilo odesílání při změně (bei Änderung). Další parametry byly nastaveny v záložce *Tastenfunktionen* (Klíčové funkce), kde se aktivovala tlačítka T1 a T2. Nastavení *Präsenzmelder* (Detektor pohybu) zůstalo v plně automatickém režimu (*Vollautomatik*). V první podzáložce detektoru pohybu byla nastavená doba vypnutí na 30 sekund. Při nastavování obou tlačítek byl vybrán režim spínání (*Betriebsart*).



Obr. 2.13: Detektor pohybu B.E.G - Indor 140-L-KNX-DX [9]

2.3 Parametrizace akčních členů

Tahle podkapitola se zaměřuje na parametrizaci použitých akčních členů umístěných v rozvaděči na zadní straně panelu.

2.3.1 ABB SA/S8.10.2.1

Tento osmikanálový spínací člen nebyl nijak parametrizován a byl ponechán ve stavu spínacího aktoru, který zasílá status pouze při změně. Úkolem tohoto aktoru je spínání LED představující topení a klimatizaci.



Obr. 2.14: Osmikanálový spínací člen ABB - SA/S8.10.2.1 [15]

2.3.2 ABB - JRA/S4.230.2.1

Jedná se o čtyřkanálový žaluziový člen, který je určen k ovládaní žaluzií. Jelikož se v projektu používají pouze 2 žaluziové okruhy, tak se využívá pouze polovina akčního členu. Využívají se první dva kanály. Jediná změna od původní parametrizace je v záložkách *Drive* pro jednotlivé kanály a to nastavení ukončení pohybu po 5 sekundách (tj. žaluzie může po stisku vyjíždět/sjíždět po dobu maximálně 5 sekund).



Obr. 2.15: Čtyřkanálový žaluziový člen ABB - JRA/S4.230.2.1 [16]

2.3.3 HDL - M/R8.10.1

Osmikanálový spínací člen HDL se v této práci využívá ke spínání osvětlení tedy 7 LED, které představují osvětlení umístěné v domě. V tomto případě nebyla nutná žádná změna oproti původnímu nastavení parametrů. Všechny kanály jsou nastaveny jako spínací aktor s typem kontaktu Normally Opened (NO). Zasílání statusu probíhá pouze při změně hodnoty.



Obr. 2.16: Osmikanálový spínací člen HDL - M/R8.10.1 [22]

2.3.4 HDL - M/DRGBW4.1

Čtyřnásobný stmívací člen poskytnutý společností HDL byl v této práci použit na ovládání RGBW LED pásku ukrytém v demonstrativním panelu. Výhodou tohoto člena je možnost ovládat kanál barevných složek zvlášť. Každý kanál (*Channel*) je nastaven na odesílání stavové hodnoty (1bit) při změně. Dále se nastavily hodnoty času pro stmívání v záložkách *dimming config* každého kanálu na 1 sekundu pro zapnutí i vypnutí.



Obr. 2.17: Čtyřnásobný stmívací člen HDL - M/DRGBW4.1 [23]

2.4 Připojená komunikační rozhraní

Pro umožnění parametrizace a externího řízení bylo nutno přidat do projektu dvě různá rozhraní pro komunikaci. První z nich je IP Secure router Siemens - 5WG1 146-1AB03, který je převážně určen k bezpečnému přenosu dat. Lze z něj také vytvořit liniovou spojku. [25]

Tomuto rozhraní byla pouze nastavena IP adresa, maska a základní brána pro komunikaci s RaspberryPi skrze tunelování.



Obr. 2.18: IP Secure router Siemens - 5WG1 146-1AB03 [25]

Druhé komunikační rozhraní použité na panelu je Weinzierl - KNX IP BAOS 774. Využívá se pro komunikaci skrze telegramy nebo datové body. Dále umožňuje přístup k objektům pomocí TCP/IP protokolu anebo za pomoci webového rozhraní. [27]

Toto rozhraní bylo také parametrizováno pro komunikaci s PLC. Byla mu nastavena odlišná IP adresa, stejná maska a brána jako u předchozího rozhraní. Dále se v něm definovaly datapointy pro komunikaci. Ty jsou definované v tabulce 2.1.



Obr. 2.19: Komunikační rozhraní Weinzierl - KNX IP BAOS 774 [28]

Tab. 2.1: Datapointy pro komunikaci KNX IP BAOS 774 s PLC

Datapoint	Typ	Účel
1.	DPT 1	Světlo pracovna zpětná vazba
2.	DPT 1	Světlo obývací pokoj zpětná vazba
3.	DPT 1	Světlo televize zpětná vazba
4.	DPT 1	Světlo linka zpětná vazba
5.	DPT 1	Světlo kuchyně zpětná vazba
6.	DPT 1	Světlo koupelna zpětná vazba
7.	DPT 1	Vchodové světlo zpětná vazba
8.	DPT 1	Topení pracovna zpětná vazba
9.	DPT 1	Topení televize zpětná vazba
10.	DPT 1	Topení kuchyň zpětná vazba
11.	DPT 1	Klimatizace pracovna zpětná vazba
12.	DPT 1	Klimatizace televize zpětná vazba
13.	DPT 1	Klimatizace kuchyně zpětná vazba
14.	DPT 1	Levé žaluzie zpětná vazba
15.	DPT 1	Levé žaluzie příkaz
16.	DPT 1	Pravé žaluzie zpětná vazba
17.	DPT 1	Pravé žaluzie příkaz
18.	DPT 1	Světlo pracovna příkaz
19.	DPT 1	Světlo obývací pokoj příkaz
20.	DPT 1	Světlo televize příkaz
21.	DPT 1	Světlo linka příkaz
22.	DPT 1	Světlo kuchyně příkaz
23.	DPT 1	Světlo koupelna příkaz
24.	DPT 1	Vchodové světlo příkaz
25.	DPT 1	Topení pracovna příkaz
26.	DPT 1	Topení televize příkaz
27.	DPT 1	Topení kuchyň příkaz
28.	DPT 1	Klimatizace pracovna příkaz
29.	DPT 1	Klimatizace televize příkaz
30.	DPT 1	Klimatizace kuchyně příkaz
31.	DPT 1	Krovování levé žaluzie
32.	DPT 1	Krovování levé žaluzie
33.	DPT 18	Scéna odchod
34.	DPT 9	Teplota okolí panelu

2.5 Vytvoření skupinových adres projektu

V závislosti na informacích obsažených v podkapitole 1.3.2 se tato podkapitola zaměří pouze na tvorbu skupinových adres. První část této podkapitoly bude věnována vytvoření a popsání tabulek jednotlivých místností. Tyto tabulky budou použity pro popis funkce jednotlivých tlačítek a následně pro tvorbu skupinových adres. Dále tyto tabulky nebudou obsahovat tlačítko společnosti Berker, které nelze parametrisovat.

První místnosti, které budou nastaveny, jsou kuchyně a koupelna. Do těchto prostor byly pomyslně nainstalovány tlačítka společnosti Ekinex a Siemens. Aby se využilo maximálně těchto tlačítek, budou využita obě tlačítka i v jiných místnostech. Zejména se jedná o tlačítko Ekinex, které má na pravé klapce žaluzie. V případě tlačítka Siemens se jedná pouze o využití velkého množství tlačítek, které budou použity při dlouhém stisku na ovládání celé budovy.

Tab. 2.2: Funkce kuchyňských tlačítek pro krátké stisknutí

Tlačítko	Ekinex	Siemens
1.	S1 Zapnuto/Vypnuto	Ch3 Vypnuto
2.	S2 Zapnuto/Vypnuto	Ch3 Zapnuto
3.	Ž1, Ž2 krok nahorů	T3 Vypnuto
4.	Ž1, Ž2 krok dolů	T3 Zapnuto
5.	-	T3/1 Vypnuto
6.	-	T3/1 Zapnuto
7.	-	T3/2 Vypnuto
8.	-	T3/2 Zapnuto

Tab. 2.3: Funkce kuchyňských tlačítek při dlouhém stisknutí

Tlačítko	Ekinex	Siemens
1.	S1, S2, S6 Zapnuto/Vypnuto	Ch1 Zapnuto/Vypnuto
2.	S6 Zapnuto/Vypnuto	Ch2 Zapnuto/Vypnuto
3.	Ž1, Ž2 nahorů	T1, T2, T2 Vypnuto
4.	Ž1, Ž2 dolů	T1, T2, T2 Zapnuto
5.	-	Ch1, Ch2, Ch3 Vypnuto
6.	-	Ch1, Ch2, Ch3 Zapnuto
7.	-	S1,S2,S6 Vypnuto
8.	-	S3,S4,S5 Zapnuto

Další místnosti se dvěma pomyslně nainstalovanými tlačítky je obývací pokoj. Jedná se o tlačítka společnosti ABB a Simon. Tlačítko Simon bude použito na ovládání žaluzí a tlačítko ABB na ovládání topení, chlazení a světel v místnosti.

Tab. 2.4: Funkce tlačítek obývacího pokoje při krátké stisknutí

Tlačítko	ABB	Simon
1.	T1 Vypnuto	Ž1 krok nahorů
2.	T2 Vypnuto	Ž1 krok dolů
3.	Ch1,Ch2 Vypnuto	Ž2 krok nahorů
4.	S3 Vypnuto	Ž2 krok dolů
5.	S4 Vypnuto	-
6.	S5 Vypnuto	-

Tab. 2.5: Funkce tlačítek obývacího pokoje při dlouhém stisknutí

Tlačítko	ABB	Simon
1.	T1 Zapnuto	Ž1 nahorů
2.	T2 Zapnuto	Ž1 dolů
3.	Ch1,Ch2 Zapnuto	Ž2 nahorů
4.	S3 Zapnuto	Ž2 dolů
5.	S4 Zapnuto	-
6.	S5 Zapnuto	-

Dotyková tlačítka společností Basalte a HDL byla určena na ovládání scén a barvy pozadí objektu. Tlačítko společnosti basalte v této práci reaguje pouze na krátký dotek jednotlivých tlačítek. Tato skutečnost je způsobena použitím scén. Při použití funkce volání scény nelze využít dlouhého doteku. Další z vlastností tlačítka je již zmiňovaný multitouch, který funguje na bázi doteku dvou a více ploch najednou. V této práci je použit krátký dotek na zavolání scény odchod a dlouhý dotek na volání scény příchod.

Tab. 2.6: Funkce dotykových tlačítek při krátké stisknutí

Tlačítko	Basalte	HDL
1.	Scéna dovolená	Červené podsvícení
2.	Scéna léto	Zelené podsvícení
3.	Scéna zima	Modré podsvícení
4.	RGB Kontroler	Bílé podsvícení
5.	-	Nastavená hodnota RGB 1
6.	-	Nastavená hodnota RGB 2

Tab. 2.7: Funkce dotykových tlačítek při dlouhém stisknutí

Tlačítko	Basalte	HDL
1.	-	Červené podsvícení stmívání
2.	-	Zelené podsvícení stmívání
3.	-	Modré podsvícení stmívání
4.	-	Bílé podsvícení stmívání
5.	-	Nastavená hodnota RGB 3
6.	-	Nastavená hodnota RGB 4

Poslední z použitých spínačů je detektor pohybu, kterému bylo logicky přiřazeno přední světlo domu.

Ze vzniklých tabulek byly vytvořeny skupinové adresy, které byly rozděleny do skupin dle přístroje (Světla, Žaluzie, Topení, Klimatizace, LED, Scény a Měření). Tyto skupiny se dále dělí podle funkcionality a množství. Poslední vrstva již představuje jednotlivé objekty nebo scény. Výpis skupinových adres je součástí příloh.

3 Vizualizace skrze programovatelný logický automat

V této kapitole se nachází popis jednotlivých částí ovládání instalace skrze PLC firmy TECO - CP-2007 [29], které obsahuje knihovny pro práci s KNX/IP [32] a MQTT [36] a dále integrovaný webový server pro vizualizaci [37]. Všechny tyto části jsou podrobněji rozvedeny v následujících podkapitolách. Ovládání a vizualizace instalace je možné provádět i skrze PLC jiných výrobců za předpokladu, že mají implementované knihovny pro komunikace KNX/IP a MQTT. PLC výrobce TECO bylo vybráno díky jeho specializaci na domácí automatizaci, dostupnosti a ceně.

3.1 CP - 2007

Jedná se o základní modul řídícího systému Foxtrot v provedení s jednojádrovým procesorem ARMv7 o frekvenci 792MHz a databoxem o velikosti 128kB, který je vyroben pro přichycení na DIN Lištu. Obsahuje 2 ethernet porty, 2 sériové porty, 15 vstupů, z nichž je 14 univerzálních a 1 galvanicky oddělený digitální, 15 výstupů, z nichž je 11 releových a 4 analogové. Dále pak obsahuje 2 sloty na rozšiřující moduly. [29]



Obr. 3.1: CP-2007 [29]

3.2 Mosaic

Ovládání instalace bylo realizováno v programovacím prostředí společnosti TECO - Mosaic, které je určeno pro programování PLC. Toto prostředí nabízí široké spektrum funkcí a nástrojů pro programování, vizualizaci a správu projektů [30]:

- **Programovací jazyky dle IEC 61131-3 [31]:**
 - Ladder Diagram (LD)
 - Function Block Diagram (FBD)
 - Structured Text (ST)
 - Instruction List (IL)
 - Sequential Function Chart (SFC)
 - Continuous Function Chart (CFC)
- **Simulační nástroje:**
 - Simulátor PLC
 - Simulátor panelu
- **Archivační nástroje:**
 - Datalogger
 - Správce souborů projektu
 - Správce knihoven
- **Nástroje pro tvorbu vizualizace:**
 - WebMaker
 - PanelMaker
 - GraphMaker
- **Inženýrské a pomocné nástroje:**
 - Mapování uživatelských registrů
 - I/O konfigurátor
 - PID Maker
 - PLCnet Manažer
 - LangMan (jazykový manažer)
 - Debugger
 - IEC Manažer
 - Asistent 16 → 32 (Převod 16bitového projektu na 32bitový)
 - Texty KEY2 (Správa textových řetězců pro operační panely KEY2)
 - Import KNX (Import konfigurace KNX/IP BAOS z csv souboru)
 - Firmware Updater
 - Project Loader
 - Set PLC IP
 - Mosaic Updater
 - Jazyk prostředí / IDE Language

3.2.1 Ovládací prvky

Pro ovládání instalace byly vytvořeny funkční bloky, které byly poté použity v realizaci logiky programu:

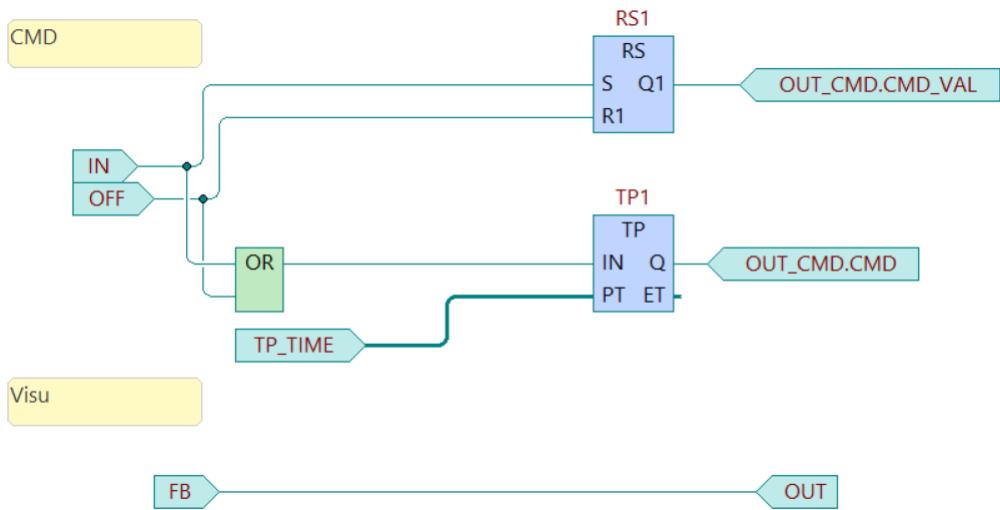
- **Základní funkční bloky:**
 - fbKNXVisuBool - Ovládání binárních signálů skrze vizualizaci
 - fbKNXShutters - Ovládání žaluzií skrze vizualizaci
 - fbRoomTempMod - Modelovaní teploty místnosti v závislosti na parametrech a vstupech z topení a klimatizace (měření teploty z panelu bude neměnné - neumožní demonstraci změny teploty)
- **Funkční bloky jednotlivých místností:**
 - fbLivRoom - Ovládání obývacího pokoje a simulace teploty
 - fbKitch - Ovládání kuchyně a simulace teploty
 - fbBath - Ovládání koupelny a simulace teploty
 - fbOutz - Ovládání vstupu a měření venkovní teploty (teplota z panelu)

Níže jsou uvedeny definice jednotlivých funkčních bloků, které byly použity pro ovládání a vizualizaci instalace. Pro jejich realizaci byly použity jazyky CFC a ST.

fbKNXVisuBool

Výpis 3.1: Definice funkčního bloku fbKNXVisuBool

```
1 FUNCTION_BLOCK fbKNXVisuBool
2   VAR_INPUT
3     IN          : BOOL R_EDGE; //Světlo ON Vizu
4     OFF         : BOOL R_EDGE; //Světlo OFF Vizu
5     FB          : BOOL; //KNX Světlo Feedback
6   END_VAR
7   VAR_OUTPUT
8     OUT_CMD    : DT_CMD_BOOL; //KNX CMD
9     OUT         : BOOL; //Visu hodnota
10  END_VAR
11  VAR
12    TP_TIME : TIME := T#1S; //KNX CMD délka
13    RS1 : RS;
14    TP1 : TP;
15  END_VAR
```



Obr. 3.2: fbKNXVisuBool

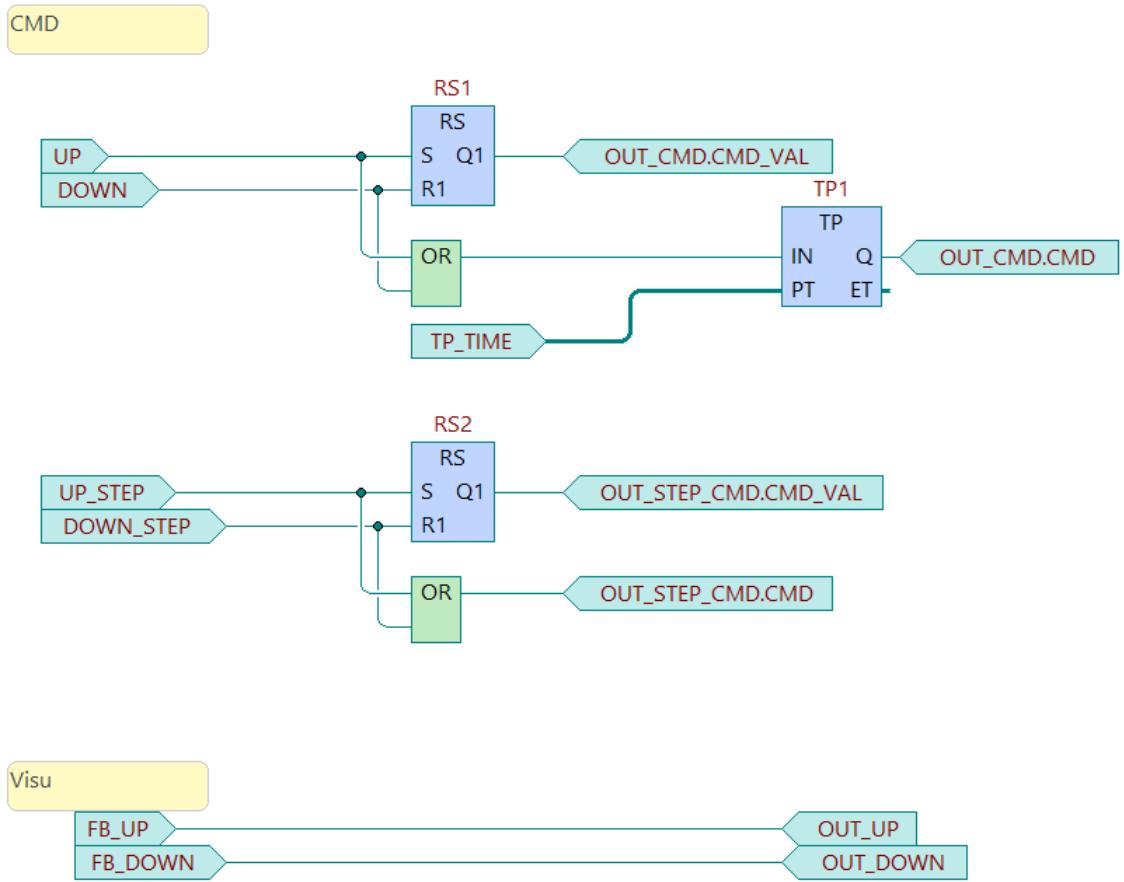
fbKNXShutters

Výpis 3.2: Definice funkčního bloku fbKNXShutters

```

1  FUNCTION_BLOCK fbKNXShutters
2  VAR_INPUT
3      UP          : BOOL R_EDGE; //Rolety nahoru Vizu
4      DOWN         : BOOL R_EDGE; //Rolety dolu Vizu
5      UP_STEP      : BOOL; //Rolety nahoru krok Vizu
6      DOWN_STEP    : BOOL; //Rolety dolu krok Vizu
7      FB_UP        : BOOL; //KNX Rolety Feedback nahoru
8      FB_DOWN       : BOOL; //KNX Rolety Feedback dolu
9  END_VAR
10  VAR_OUTPUT
11     OUT_CMD      : DT_CMD_BOOL; //KNX CMD
12     OUT_STEP_CMD : DT_CMD_BOOL; //KNX CMD krok
13     OUT_UP        : BOOL; //Visu nahoru
14     OUT_DOWN      : BOOL; //Visu dolu
15  END_VAR
16  VAR
17     TP_TIME : TIME := T#1S; //KNX CMD délka
18     RS1 : RS;
19     TP1 : TP;
20     RS2 : RS;
21  END_VAR

```



Obr. 3.3: fbKNXShutters

fbRoomTempMod

Na modelování teploty místnosti byl vytvořen jednoduchý matematický model, který měl za úkol zobrazit změnu v závislosti na působení topení a klimatizace. Uvažujeme, že místnost je kvádr o rozměrech specifikovaných uživatelským vstupem. Tento kvádr je naplněný vzduchem, který má určitou váhu, ze které lze vypočítat energii potřebnou ke změně teploty o 1 °C.

$$V = a \cdot b \cdot c \quad (3.1)$$

$$m = \rho \cdot V \quad (3.2)$$

$$Q = m \cdot c_p \quad (3.3)$$

Kde:

- V – objem vzduchu v místnosti [m^3]
- a, b, c – rozměry místnosti [m]
- m – hmotnost vzduchu v místnosti [kg]

- ρ – hustota vzduchu při tlaku jedné atmosféry a teplotě 20 °C - 1.204 [kg/m³]
- Q – energie potřebná ke změně teploty o 1 stupeň teploty [J]
- c_p – měrná tepelná kapacita vzduchu - 1005 [J·kg⁻¹·K⁻¹]

Dále uvažujeme obal kvádru (stěny, podlahu a strop) s různými tloušťkami a tepelnými vodivostmi (Fourierův zákon vedení tepla 3.6). Pro zjednodušení výpočtu se předpokládá, že dveře a okna místnosti nemají rozdílný vliv oproti stěnám, tudíž tepelný tok zůstává na celé ploše strany stejný. Zároveň směr toku závisí pouze na poměru teplot na obou stranách zdi. Dále je předpoklad nulových teplot z jiných směrů. Také budeme předpokládat, že materiál bude všude stejný a to pálená cihla.

$$S = a \cdot b \quad (3.4)$$

$$\Delta T = T_{in} - T_{out} \quad (3.5)$$

$$\phi_{Strana} = \frac{\lambda \cdot S \cdot \Delta T}{c} \quad (3.6)$$

Kde:

- S – plocha strany [m²]
- a, b, c – rozměry strany [m]
- ΔT – rozdíl teploty na obou stranách strany [°C]
- T_{in} – teplota uvnitř místnosti [°C]
- T_{out} – teplota venku [°C]
- ϕ_{Strana} – tepelný tok [W]
- λ – tepelná vodivost cihly - 0.4 [W·m⁻¹·K⁻¹]

Další součástí simulace je topení a klimatizace, které jsou dodávány pouze jako binární signály. Pro jednoduché modelování byly vytvořeny funkce růstu (logaritmický) a poklesu (exponenciální) výkonu. Tyto funkce jsou zobrazeny na Obr. 3.4. Dále byly k těmto rovnicím vytvořeny korekční členy, které ovlivňují rychlosť funkcí, a tedy více přiblížily chování funkcí realitě.

$$f_{Růst}(t) = \frac{\ln(1 + k_{Růst} \cdot t)}{\ln(1 + k_{Růst} \cdot t_{MaxRůst})} \cdot y_{Max} \quad (3.7)$$

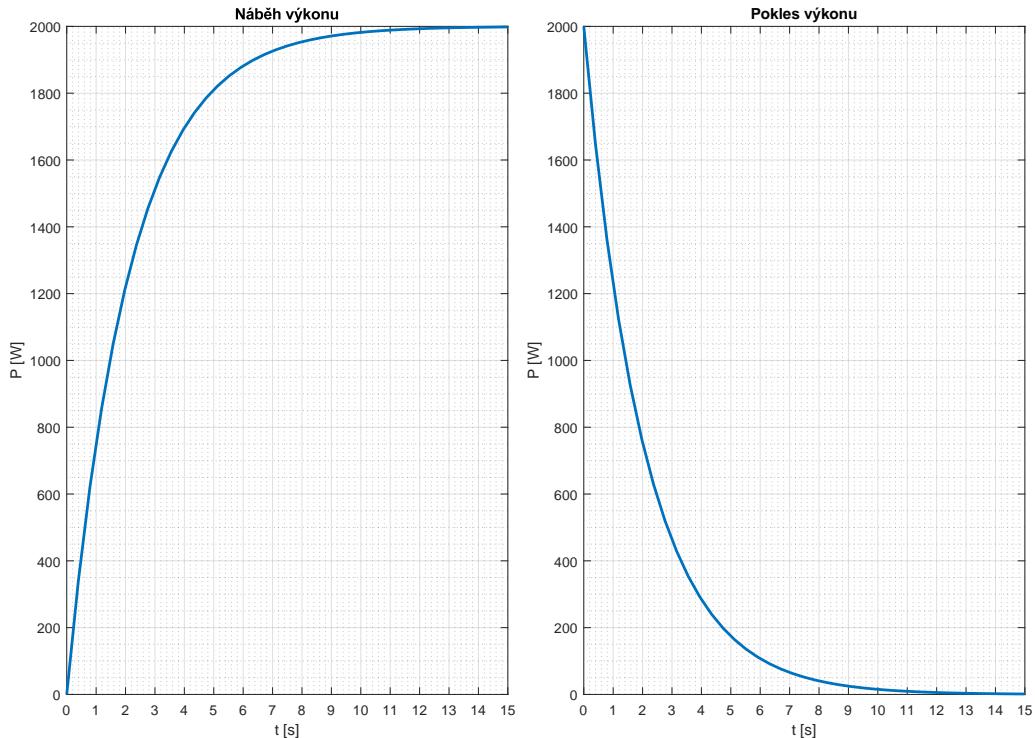
$$k_{Růst} = \frac{t_{MaxRise}}{t_{80}} \quad (3.8)$$

$$f_{Pokles}(t) = \frac{y}{e^{k_{Pokles} \cdot t}} \quad (3.9)$$

$$k_{Pokles} = \frac{\frac{y_{max}}{\epsilon}}{t_{MaxPokles}} \quad (3.10)$$

Kde:

- $f_{Růst}(x)$ – funkce růstu výkonu [W]
- $k_{Růst}$ – korekční člen pro pokles [-]
- t – čas [s]
- $t_{MaxRůst}$ – maximální čas růstu výkonu [s]
- y_{Max} – maximální výkon [W]
- t_{80} – čas potřebný k dosažení 80% výkonu [s]
- $f_{Pokles}(x)$ – funkce poklesu výkonu [W]
- k_{Pokles} – korekční člen pro růst [-]
- y – aktuální výkon [W]
- ε – cílová hodnota, pod kterou se výkon musí dostat [W]
- $t_{MaxPokles}$ – maximální čas poklesu výkonu [s]



Obr. 3.4: Průběh funkcí pro simulaci

Tyto funkce bylo potřeba převést do rekurzivní podoby, aby bylo možné je implementovat do funkčního bloku.

$$y_{RůstTed} = y_{RůstPred} + \frac{\ln(1 + k_{Růst} \cdot \Delta t)}{\ln(1 + k_{Růst} \cdot T_{Max})} \cdot (y_{Max} - y_{RůstPred}) \quad (3.11)$$

$$y_{PoklesTed} = y_{PoklesPred} \cdot e^{k_{Pokles} \cdot \Delta t} \quad (3.12)$$

Kde:

- $y_{RustTed}$ – nový výkon [W]
- $y_{RustPred}$ – předchozí výkon [W]
- k_{Rust} – korekční člen pro pokles [-]
- Δt – časový krok [s]
- T_{Max} – maximální čas růstu výkonu [s]
- y_{Max} – maximální výkon [W]
- $y_{PoklesTed}$ – nový výkon [W]
- $y_{PoklesPred}$ – předchozí výkon [W]

Po vyčtení výkonu topení a klimatizace zbývá vypočítat teplotu v místnosti. Ta se získá součtem aktuální hodnoty teploty, přírůstku podílu působících toků a potřebné energie ke změně teploty. (Rov. 3.3). Rovnice pro celkový tepelný tok je dána jako součet tepelných toků ze stran, výkonu topení a klimatizace (působí záporně).

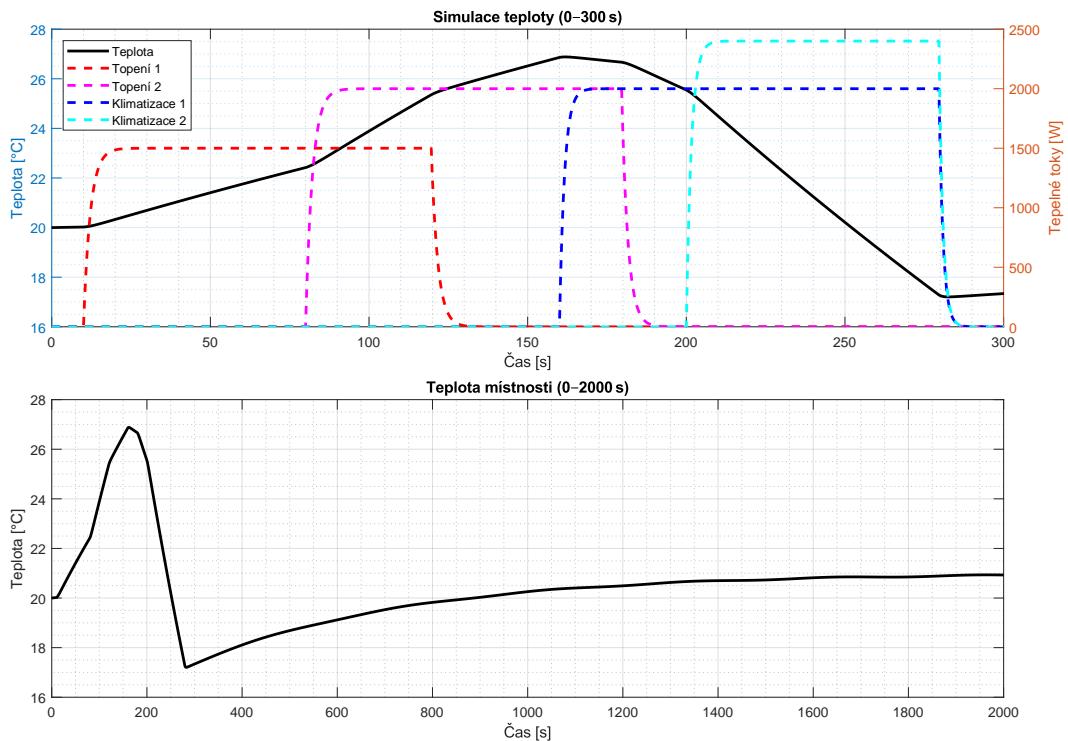
$$T_{Aktualni} = T_{Predchozi} + \frac{\phi_{Celkovy} \cdot \Delta t}{Q} \quad (3.13)$$

$$\phi_{Celkovy} = \sum_{i=1}^6 \phi_{Strana_i} + \phi_{Topeni} - \phi_{Klimatizace} \quad (3.14)$$

Kde:

- $T_{Aktualni}$ – aktuální teplota [$^{\circ}\text{C}$]
- $T_{Predchozi}$ – předchozí teplota [$^{\circ}\text{C}$]
- Δt – časový krok [s]
- $\phi_{Celkovy}$ – celkový tepelný tok [W]
- ϕ_{Strana_i} – tepelný tok ze strany i [W]
- ϕ_{Topeni} – tepelný tok z topení [W]
- $\phi_{Klimatizace}$ – tepelný tok z klimatizace [W]

Průběh teploty v místnosti je zobrazen na Obr. 3.5, kde je demonstrováno chování při běhu topení a klimatizace v horním grafu. Ve spodní části grafu je zase zobrazeno chování teploty bez působení topení a klimatizace. Teplota okolí je v tomto případě nastavená na $21\ ^{\circ}\text{C}$. Z průběhu lze vypozorovat, že se teplota ustálí na teplotu okolí a tudíž, lze považovat model za korektní.



Obr. 3.5: Simulace teploty v místnosti

Tento funkční blok byl použit pro simulaci teploty v obývacím pokoji, kuchyni. Definice tohoto funkčního bloku je zobrazena kvůli své velikosti v příloze C.0.1. Pro venkovní teplotu byl použit signál, který byl napojen na teploměr na panelu. V případě koupelny byla teplota simulována jako sinusoida v rozmezí 18–22 °C s periodou 3 minuty. Tento způsob byl zvolen kvůli absenci teploměru a akčního členu pro řízení teploty.

funkční bloky jednotlivých místností

Funkční bloky jednotlivých místností se skládají z výše uvedených funkcí, které byly poskládány v závislosti na požadavcích jednotlivých místností - viz Obr. 2.3. Všechny funkční bloky mají definici vypsanou v přílohách dokumentu:

- **fbBath** - V příloze C.0.2
- **fbKitch** - V příloze C.0.3
- **fbLivRoom** - V příloze C.0.4
- **fbOutz** - V příloze C.0.5

3.2.2 Komunikace KNX/IP

Důležitou součástí ovládání instalace je komunikace s KNX/IP, která byla realizována pomocí knihovny KNXlib [32]. Obsahem této knihovny jsou dva funkční bloky určené ke komunikaci:

- **fbKnxIpBaos** - Funkční blok pro komunikaci s KNX/IP BAOS 772 v režimu TCP master
- **fbKnxIpBaosBin** - Funkční blok pro komunikaci s KNX/IP BAOS 772/774 skrze binární protokol v režimu TCP Master

Dále tato knihovna obsahuje datové struktury, které obsahují definice jednotlivých datapointů, které jsou potřeba pro komunikaci mezi PLC a KNX/IP Baos 774:

Tab. 3.1: Definice KNX datapointů v PLC

Typ objektu KNX	Datový typ PLC	Popis
DPT 01	T_KNX_OBJECT_DPT1	Binární signál – 1 Bit
DPT 02	T_KNX_OBJECT_DPT2	Binární signál s kontrolou (Priorita) – 2 Bity
DPT 03	T_KNX_OBJECT_DPT3	Stmívání nahoru/dolů – 4 Bit
DPT 04	T_KNX_OBJECT_DPT4	Znak – 1 bajt
DPT 05	T_KNX_OBJECT_DPT5	Škálování – 1 bajt (8 Bitová hodnota)
DPT 06	T_KNX_OBJECT_DPT6	Hodnota se znaménkem – 1 bajt
DPT 07	T_KNX_OBJECT_DPT7	Hodnota bez znaménka – 2 bajty
DPT 08	T_KNX_OBJECT_DPT8	Hodnota se znaménkem – 2 bajty
DPT 09	T_KNX_OBJECT_DPT9	Hodnota s desetinnou čárkou – 2 bajty
DPT 10	T_KNX_OBJECT_DPT10	Čas – 3 bajty
DPT 11	T_KNX_OBJECT_DPT11	Datum – 3 bajty
DPT 12	T_KNX_OBJECT_DPT12	Hodnota bez znaménka – 4 bajty
DPT 13	T_KNX_OBJECT_DPT13	Hodnota se znaménkem – 4 bajty
DPT 14	T_KNX_OBJECT_DPT14	Hodnota s desetinnou čárkou – 4 bajty
DPT 15	T_KNX_OBJECT_DPT15	Přístupová data – 4 bajty
DPT 16	T_KNX_OBJECT_DPT16	Řetězec znaků – 14 bajty
DPT 17	T_KNX_OBJECT_DPT17	Scéna – 1 bajt
DPT 18	T_KNX_OBJECT_DPT18	Kontroler scén – 1 bajt
DPT unknown	T_KNX_OBJECT_RAW	Max. 14 bajtů, Data jako pole, byte-by-byte

Realizace komunikace KNX/IP

Pro realizaci v tomto případě bylo potřeba kromě přidání knihovny a vytvoření programu ještě i nastavit ethernet port. V tomto případě byl do *ETH2* přidán obecný kanál *UNI1*, který bylo potřeba nastavit jako TCP Server. Dále je pak nutné

vytvořit několik proměnných dle tabulky datapointů 2.1, získat jejich adresy a poté je vložit do pole.

Dalším krokem je nastavení funkčního bloku *fbKnxIpBaosBin*. Nejprve se nastaví číslo prvního a posledního datapointu, který má funkční blok zpracovat, poté se nastaví použitý ethernet port a IP adresa KNX/IP BAOS 774 instalace. Nakonec se předá do portu *knxList* námi definované pole s adresami. Po tomto kroku je už možné pracovat s proměnnými a vytvoření ovládací logiku.

V případě této práce se používají pouze 3 druhy datapointů a to:

- **DPT 1** - ovládání a indikace binárního signálu
- **DPT 9** - hodnota teploty z teploměru
- **DPT 18** - ovládání scény

Níže je uvedena ukázka realizace komunikace, která vyplývá z implementace uvedené v příloze D:

Výpis 3.3: Implementace komunikace KNX/IP

```

1 PROGRAM prgKNXComm
2
3     VAR
4
5         init : BOOL;
6
7         knx : fbKnxIpBaosBin;
8
9         knxObjectList : ARRAY[1..4] OF UDINT; // pole adres
10
11        datapoint1 : T_KNX_OBJECT_DPT1;      // SV1_FB
12        datapoint2 : T_KNX_OBJECT_DPT1;      // SV1_CMD
13        datapoint3 : T_KNX_OBJECT_DPT18;     // scéna
14        datapoint4 : T_KNX_OBJECT_DPT9;      // teplota
15
16    END_VAR
17
18 IF NOT init THEN // Pole adres
19
20     knxObjectList[1] := PTR_TO_UDINT( ADR(datapoint1));
21     knxObjectList[2] := PTR_TO_UDINT( ADR(datapoint2));
22     knxObjectList[3] := PTR_TO_UDINT( ADR(datapoint3));
23     knxObjectList[4] := PTR_TO_UDINT( ADR(datapoint4));
24     init := TRUE;
25
26 END_IF
27
28 knx( firstKnxObject := 1,
29       lastKnxObject := 4,
30       ethCode := ETH2_uni2,
31       knxIP := STRING_TO_IPADR('192.168.xxx.xxx'),
32       knxList := void( knxObjectList));

```

```

25
26 SV1_FB := datapoint1.value; // Feedback
27 IF SV1_CMD THEN datapoint2.value := SV1_CMD.CMD_VAL; // CMD
28
29 IF SCENE THEN // scéna
30   datapoint3.control := TRUE;
31   datapoint3.scene    := 5;
32 END_IF
33
34 KNX_TEMPER := datapoint4.value;// Posílání teploty
35 END_PROGRAM

```

3.2.3 Komunikace MQTT

Tato sekce je zaměřená na realizaci komunikace skrze protokol MQTT, který je navržen jako lehký přenos zpráv mezi zařízeními a aplikacemi. Jeho cílovou skupinou jsou zařízení s malou kódovou stopou a minimální šírkou pásma sítě. [33]

Protokol

Základními myšlenkami protokolu MQTT je posílání zpráv mezi vydavatelem (publisher) a odběratelem(client) skrze zprostředkovatele (broker). Tento broker zajistuje, že zprávy jsou doručeny správným odběratelům. Vydavatel a odběratel nemusí být navzájem známi a komunikují pouze skrze brokera. [33]

Tento druh komunikace je realizován pomocí packetů (balíků dat) posílaných mezi zařízeními. Jednotlivé datové balíky jsou tvořen třemi částmi [34]:

- **Fixní hlavičku** - obsahuje informace o typu packetu, příznaky a velikost zbylé části packetu
- **Volitelnou hlavičku** - obsahuje dynamické informace o packetu
- **Datový obsah** - obsahuje data, která se posílají mezi zařízeními

Zprávy se poté posílají na určité téma (topic), které je definováno jako hierarchická struktura. Tato struktura je tvořena jednotlivými úrovněmi oddělenými lomítkem. V případě této práce je dobrým příkladem téma *plc/Connect/Publisher*, kde:

- **plc** - označuje zařízení, které zprávu odesílá
- **Connect** - která část systému zprávu odesílá
- **Publisher** - bližší rozlišení

Dálší část je text samotné zprávy, která je odeslána na dané téma. Tato zpráva může mít libovolnou strukturu i velikost(maximálně 256 MB). Tělem zprávy je text ve formátu JSON, který se lze pochopit jako hierarchická struktura dat. Vertikálně je struktura rozdělena do úrovní pomocí vnořených objektů a polí. Horizontálně je struktura členěna do jednotlivých klíč:hodnota párů na jedné úrovni, kde každý klíč reprezentuje konkrétní parametr. Příkladem takové zprávy je:

Výpis 3.4: Příklad zprávy v JSON

```

1 {
2     "device": "plc",
3     "location": {
4         "room": "kitchen",
5         "floor": 1
6     },
7     "sensors": {
8         "temperature": {
9             "value": 22.5,
10            "unit": "C"
11        },
12        "humidity": {
13            "value": 45,
14            "unit": "%"
15        }
16    },
17    "status": "online",
18    "timestamp": "2024-06-01T12:34:56Z"
19 }
```

Součástí zprávy je i kvalita služeb *QoS* (Quality of Service), která určuje úroveň spolehlivosti doručení zprávy. Tato úroveň je nastavena na 0 (doručení maximálně jednou bez záruky), 1(doručená alespoň jednou) nebo 2(doručená přesně jednou s potvrzením příjemce) [34]. V případě této práce byla zvolena úroveň QoS 1, která je dostatečná pro většinu aplikací a poskytuje dobrou rovnováhu mezi spolehlivostí a výkonem.

Poslední součástí zprávy je *Last Will and Testament* (LWT), což je zpráva, která je odeslána brokerem v případě, že se zařízení odpojí bez předchozího oznámení. Tato zpráva může být užitečná pro monitorování stavu zařízení a detekci výpadků [?].

MQTTlib

Pro realizaci komunikace mezi PLC a Home Assistantem byla použita knihovna MQTTlib, která obsahuje funkční bloky pro nastolení komunikace - *fbMQTTPub-*

lisher, fbMQTTSubscriber. [36]

Prvním krokem pro realizaci komunikace je nastavení ethernet portu, který bude použit pro komunikaci. V případě této práce byl použit stejný port jako pro KNX/IP komunikaci (*ETH2*). Na tomto portu byly vytvořeny dva obecné kanály *UNI0* a *UNI1*, které byly nastaveny jako TCP klienty s šíkou pásma 512 bytů. Počet kanálu závisí na počtu nastolených spojení. Pro jednoduchost se jedná o jedno spojení pro publikování a jedno pro odběr zpráv.

Dalším krokem je nastavení funkčních bloků. V obou případech je nastavení obdobné. Nejprve se nastaví kanál na kterém bude komunikace probíhat - IP adresa brokeru, port brokeru, lokální port, časový interval pro kontrolu spojení a délku nečinnosti. Dále se nastaví parametry komunikace a poslední vůle:

- **clientID_auto** - automatické generování identifikátoru klienta
- **clientID** - identifikátor klienta
- **loginName** - uživatelské jméno pro připojení k brokeru
- **loginPass** - heslo pro připojení k brokeru
- **retain** - určuje zda má být zpráva uchována na brokeru
- **QoS** - určuje úroveň kvality služeb
- **cleanSession** - určuje zda má broker uložit stav relace po výpadku
- **topic** - téma komunikace
- **mess** - zpráva komunikace
- **flag** - určuje zda má být poslední vůle odeslána v případě odpojení klienta bez předchozího oznámení

Po nastavení funkčních bloků je potřeba definovat proměnné na uchovávání zpráv. Maximální délka zprávy je 256 znaků pro oba funkční bloky. Níže je uveden příklad nastavení komunikace pro publikování zprávy:

Výpis 3.5: Příklad nastavení komunikace pro publikování zprávy

```
1 PROGRAM prgMQTTPub
2     VAR
3         mqttPub : fbMQTTPublisher;
4
5         brokerIPAddr      : STRING := '192.168.xxx.xxx'; // IP adresa
6         ↪ brokeru
7             brokerPort      : UINT := 1883; // Port brokeru
8             localPort        : UINT := 60000; // Lokální port
9             keepAlive        : BOOL := TRUE; // Udržení spojení po vypadku
10            keepAliveInterval : TIME := T#60s; // Konec spojení bez dat
11            pingInterval    : TIME := T#10s; // Kontrola spojení
```

```

11      connTimeOut      : TIME   := T#5s; // Maximalni delka odesvy
12
13      pubComParam     : T_MQTT_COM_PUB_PARAM := (
14          pRetain := TRUE, // Uchovani zpravy na brokeru
15          qos     := 1, // Kvalita sluzeb
16          dup     := FALSE, // Opakovani zpravy
17          clean   := FALSE // Uchovani stavu relace
18      );
19
20      willParamPub   : T_MQTT_COM_WILL_PARAM := (
21          wRetain := FALSE, // Uchovani posledni vule na brokeru
22          topic   := 'plc/Connect/Publisher', // Tema posledni vule
23          mess    := 'Disconnected', // Zprava posledni vule
24          flag    := TRUE, // Odeslani posledni vule
25          qos     := 1 // Kvalita sluzeb
26      );
27
28      jsonPayload : STRING[255]; // Zprava
29      topic      : STRING[80] := 'plc/test'; // Tema
30  END_VAR
31
32 jsonPayload := CONCAT(
33     '{"test"}'
34 );
35 mqttPub(
36     chanCode      := ETH2_uni1, // Fyzicky kanal
37     brokerIP      := STRING_TO_IPADDR(brokerIPAddr),
38     brokerPort    := brokerPort,
39     localPort     := localPort,
40     connect       := TRUE, // Pripojit k brokeru
41     keepAlive     := keepAlive,
42     keepAliveInterval := keepAliveInterval,
43     pingInterval  := pingInterval,
44     connTimeOut   := connTimeOut,
45     clientId_auto := FALSE, // Automaticke generovani ID
46     clientId      := 'TEST_PUB', // ID klienta
47     comParam      := pubComParam, // Parametry komunikace
48     willParam     := willParamPub, // Parametry posledni vule
49     topicTxt      := topic, // Tema

```

```

50     dataTxt          := jsonPayload, // Zprava
51     sendCom         := TRUE // Odeslani zpravy
52 );
53 END_PROGRAM

```

Implementace použitá v této práci je uvedena v příloze E. Jedná se o nastavení komunikace pro komunikaci s Home Assistantem, který přijímá hodnoty teplot místností a odesílá příkaz pro celkové vypnutí instalace.

3.2.4 Vizualizace

V této sekci je popsána tvorba vizualizace panelu pomocí integrovaného webového serveru. Nastavení a realizace webového serveru je realizována pomocí integrovaného nástroje WebMaker. Tento nástroj je popsán v dokumentaci [37].

Prvním krokem je vytvoření struktury projektu. V tomto případě to bude 5 stránek ve formátu .XML zasazených do webové struktury:

```

/Vizualizace
└── Prehled - PAGE1.xml
└── Veranda - PAGE2.xml
└── Kuchyn - PAGE3.xml
└── Koupelna - PAGE4.xml
└── Obyvaci_pokoj - PAGE5.xml

```

Druhým krokem bylo importování sady ikon, které byly použity jako vícerozměrné obrázky napojené na výstupy funkčních bloků uvedených v podkapitole 3.2.1. Tyto ikony byly získány na stránce Flaticon [38], která poskytuje vektorové ikony na projekty zdarma s možností úpravy.



Obr. 3.6: Ikony použité pro vizualizaci

Dále se musely vytvořit ovládací tlačítka jednotlivých objektů. Tato tlačítka byla také vytvořena jako vícerozměrné obrázky, které byly napojeny na vstupy funkčních bloků. Pro zobrazování teplot v jednotlivých místnostech byla použita zadávací pole s parametrem *Read only*. Výsledná vizualizace je uvedena v příloze F. Vzhled byl zvolen minimalistický a funkční s ohledem na viditelnost a přehlednost. Černé pozadí zdůrazňuje kontrast ikon a textu. Dalším důvodem byla ochrana zraku při ovládání v noci.

4 Open-source platformy

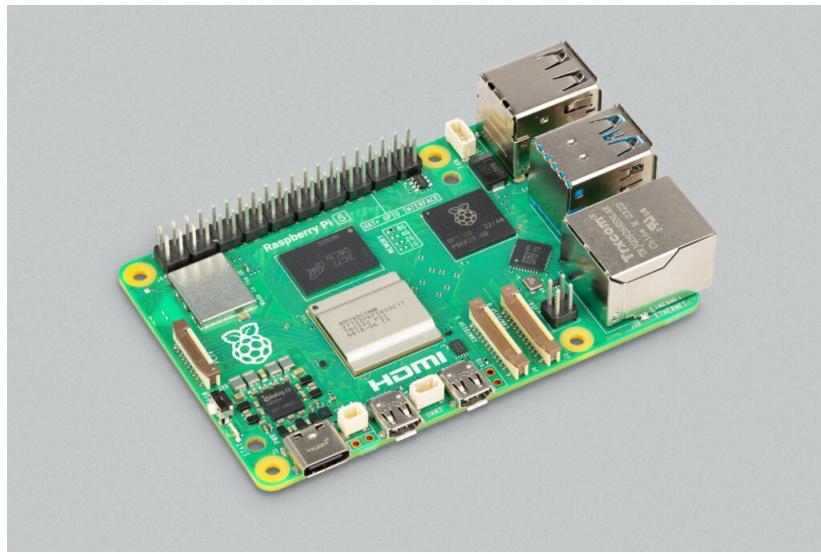
Tato kapitola se zaměřuje na možnost vizualizace a ovládní domácí automatizace pomocí open-source technologií (software s otevřeným kódem), které je možné využít v případě nedostatku financí nebo pro uživatele, kteří chtějí mít plnou kontrolu nad svým systémem. V případě tohoto řešení byl vybrán přístup pomocí kontejnerizace, který umožňuje snadné nasazení a škálování aplikací, ale také snadné zálohování a migraci v případě poruch.

4.1 Raspberry Pi 5

Jedná se o jednodeskový počítač, který je určen pro široké spektrum aplikací od IoT, rozpoznávání obrazu, strojového učení, robotiky až po multimediální aplikace. Mezi hlavní výhody patří nízká cena, malé provozní náklady, možná modularita přes GPIO piny, PCI expres a USB porty a velké množství tzv. HAT modulů, které jsou určeny pro rozšíření funkcí. [39].

Pro tuto aplikaci bylo zvoleno Raspberry Pi 5 s 16 GB RAM, pamětí 256 GB a operačním systémem Raspberry Pi OS, který je stavěn na Linuxové distribuci Debianu. Hlavním důvodem výběru byla nízká cena, ale i jednoduchá dostupnost pro případného zájemce o domácí automatizaci, který by si chtěl systém ovládání a vizualizace vytvořit sám.

Jediným závažnějším problémem tohoto zařízení je degradace paměťové karty, na které je nahrán operační systém. Tento problém je způsoben častým zápisem, který postupně sniže životnost, rychlosť a velikost paměti. V krajních případech může dojít i k úplnému zničení a ztrátě dat. K předcházení tohoto problému se historicky používala technika *wear leveling* - rozložení zápisu na celou paměťovou kartu, aby se snížil počet zápisů na jednotlivé buňky [40]. Teď už je možné použít M.2 NVMe SSD disk, který je připojen přes PCI expres a je mnohem rychlejší než paměťová karta. Tento disk je možné použít i jako bootovací disk, což zamezí problémům s bootováním a ztrátou dat. Další možnosti, jak zpomalit tuto degradaci, je připojení USB flash disku. Flash disk dokáže zamezit opotřebování paměti díky přenesení souborů s vyšší frekvencí zápisu - v tomto případě databáze, nicméně rychlosť zápisu se tímto sníží.



Obr. 4.1: Raspberry Pi5 [39]

Alternativou k Raspberry Pi 5 mohou být LattePanda, ASUS NUC a podobné mikropočítače, které mají vyšší výkon, větší možnost výběru operačního systému (dostatečný výkon na Windows) a robustnost. Tyto výhody jsou kompenzovány vyšší pořizovací cenou, spotřebou a velikostí.

4.2 Docker Compose

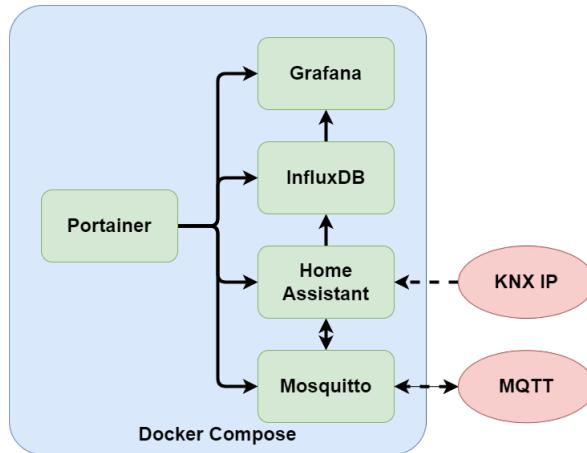
Docker Compose je nástroj pro definici a správu více kontejnerových aplikací. Umožňuje uživatelům definovat aplikaci pomocí *YAML* souboru, které obsahují informace o kontejnerových službách, sítích a úložištích. [41]

Instalace Docker Compose je jednoduchá a rozepsaná na oficiálních stránkách Dockeru. Obsahuje pouze tři kroky, které jsou již připraveny pro kopírování do terminálu. [42]

4.2.1 Kontejnerizace

Kontejnerizace je způsob virtualizace aplikací založený na linuxové technologii LXC (Linux Containers), která umožňuje spouštět aplikace v izolovaném prostředí se zárukou jejich funkčnosti v různých systémech. Tento přístup je výhodný pro nasazení aplikací s různými závislostmi a konfiguracemi a umožňuje jejich snadné nasazení i škálování. Na rozdíl od tradičních virtuálních strojů poskytuje kontejnerizace kompletní běhové prostředí s menšími nároky na výkon a paměť, protože nevyžaduje samostatný kernel ani simulaci veškerého hardwaru. [43] V případě této práce byl

navrhnut Docker Stack, který je složen z několika kontejnerů, které spolu komunikují a vytváří tak komplexní systém na ovládání a vizualizaci domácí automatizace. Vzhled tohoto stacku i s komunikacemi je zobrazen na Obr. 4.2.



Obr. 4.2: Docker Stack

4.2.2 Tvorba YAML souboru

YAML soubor je textový soubor, který obsahuje definici kontejnerů a jejich konfigurací. Je napsán v jazyce YAML (YAML Ain't Markup Language), což je formát pro serializaci dat [44]. Níže je příklad vzhledu YAML souboru (Výpis 4.1), který by měl pomoci ke snadnému pochopení struktury a syntaxe [45].

Dále pak může YAML soubor obsahovat i enviromentální proměnné, které se používají k nastavení kontejneru. Tyto proměnné mohou být obsaženy v souboru .env, který se používá k uchovávání citlivých informací, jako jsou například hesla a API klíče. Tento soubor je načítán při spuštění kontejneru. [46]

Výpis 4.1: YAML soubor

```

1 version: "3" # Verze Docker Compose
2 services: # Sluzby
3     name_of_service: # Jmeno sluzby
4         image: name_of_image:latest # Jmeno image
5         container_name: name_of_container # Jmeno kontejneru
6         networks: # Site
7             - name_of_network # Jmeno site na ktore pobezi kontejner
8         depends_on: # Zavislosti
9             - name_of_service # Jmeno kontejneru na kterem zavisi

```

```

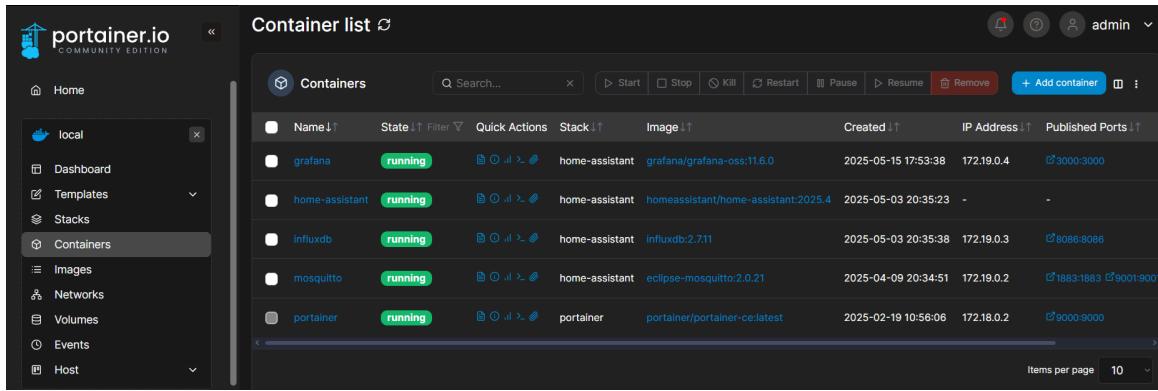
10    environment: # Promenne prostredi
11        - PUID=1000 # ID uzivatele, ktery bude mit pristup k souborum
12        - PGID=1000 # ID skupiny, ktera bude mit pristup k souborum
13        - TZ=Europe/Prague # Casove pasmo
14        - WEBUI_PORT=1234 # Port na kterem pobezi webove rozhrani
15        - DOCKER_MODS="linuxserver/mods:universal" # Docker modifikace
16    volumes: # Slozky, ktere budou pripojeny do kontejneru
17        - "/path/on/host:/path/in/container" # Cesta k souborum
18    ports: # Porty pro pripojeni k hostitelske site
19        - "host_port:container_port"
20    deploy: # Nasazeni kontejneru
21        resources:
22            limits:
23                memory: 512m # Maximalni pamet
24                cpus: "1" # Maximalni CPU
25            reservations:
26                memory: 256m # Minimalni pamet
27                cpus: "0.5" # Minimalni CPU
28    restart: always # Restart kontejneru pri padu
29    labels: # Metada
30        - "com.docker.compose.project=project_name" # Nazev projektu
31        - "com.docker.compose.service=service_name" # Nazev sluzby
32        - "com.docker.compose.version=1.0" # Verze sluzby

```

Celková implementace a je zobrazena v příloze G.

4.2.3 Portainer

Jedná se o open-source kontejner, který slouží jako webové rozhraní pro správu Dockeru. Umožňuje uživatelům spravovat kontejnery, image (obrazy – šablony pro vytváření kontejnerů), stohy (stohy – skupiny kontejnerů, které spolu komunikují), síť, uložiště, čtení logů, sledování výkonu, správa portů a další funkce. Jednou z předních výhod je jednoduchost použití; kromě rozhraní (Obr. 4.3) je i propojení s Docker Hubem, což je veřejná knihovna kontejnerů. [47]



Obr. 4.3: Portainer Stack

4.3 Mosquitto

Mosquitto je open-source broker společnosti Eclipse pro protokol MQTT (Message Queuing Telemetry Transport – více v podkapitole 3.2.3). Zajišťuje komunikaci mezi vydavatelem a odběratelem v závislosti na oprávněních a kvalitě služeb (*QoS*). Jednou z výhod tohoto brokeru je jednoduchost implementace a nízké nároky na výkon. Dále umožňuje šifrování pomocí *TLS*, které zajišťuje bezpečnost přenosu dat, podporuje cloudové služby a různé platformy. K dispozici je také knihovna pro jazyk *C*, která implementuje celý protokol. [48]

V této práci byl použit kontejner pro komunikaci mezi Home Assistantem a PLC. Tento kontejner byl nastaven na porty 1883 a 9001. Port 1883 je standardní port pro MQTT protokol a používá se ke komunikaci mezi brokerem a PLC. Port 9001 je určen pro WebSocket, což je protokol pro komunikaci mezi webovými aplikacemi a servery. Proto byl použit pro komunikaci mezi Home Assistantem a Mosquittem.

Různé porty v brokeru nemají vliv na funkčnost komunikace, protože broker je schopen komunikovat na více portech současně. To znamená, že broker může přijímat zprávy na jednom portu a odesílat je skrze jiný port, pokud je použito správné téma (*topic*). Dále nezáleží na pořadí příchodu zpráv, protože broker je zpracovává sekvenčně – v pořadí, v jakém přišly.

4.4 Home Assistant

Home Assistant je open-source platforma pro domácí automatizaci, která v posledních letech získala velkou popularitu. Umožňuje propojení více zařízení různých výrobců, databází a protokolů do jednoho systému. Dále pak umožňuje uživatelům monitorovat spotřebu, vytvářet vlastní vizualizace, automatizace a scénáře pro ovládání zařízení. Tato platforma je dostupná v kontejnerové podobě, jako operační

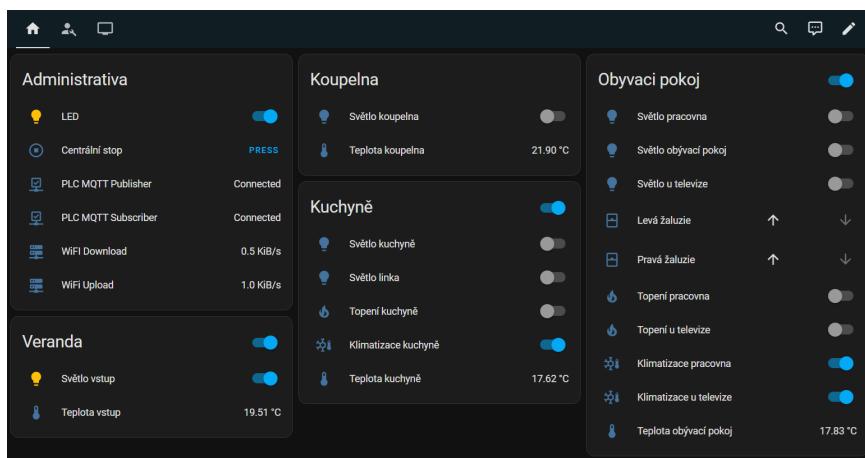
systém nebo jako virtuální stroj. [49]

Jednotlivá zařízení lze implementovat prostřednictvím tzv. integrací, které jsou buď předinstalované v Home Assistantu, nebo dostupné jako pluginy. Portfolio zařízení i funkcí lze dále rozšířit pomocí komponent vyvíjených komunitou, které jsou dostupné prostřednictvím pluginu *HACS* (Home Assistant Community Store). [50]

Přidání KNX zařízení do Home Assistantu je možné pomocí integrace *KNX*, která umožňuje implementaci skrze YAML soubor, ale také pomocí webového rozhraní. V této práci byla zvolena druhá možnost, která je jednodušší a rychlejší. Prvním krokem byl výběr a nastavení spojení s KNX - *routing*, **tunneling** (více v podkapitole 1.3.4). Dalším krokem bylo vložení projektu instalace vygenerovaného pomocí *ETS*, který obsahoval všechny potřebné informace o zařízeních a jejich adresách. Posledním krokem byla tvorba jednotlivých entit, které se vkládaly pomocí webového rozhraní - výběr typu entity, její název a adres jednotlivých funkcí. Například pro světlo bylo potřeba nastavit adresu pro ovládání, adresu pro stav, adresu pro jas a adresy pro nastavení barev. Tyto adresy se pak vybíraly ze seznamu, který se vygeneroval při vložení projektu. Existuje také možnost použít i vlastní adresy, které v projektu dosud neexistují.

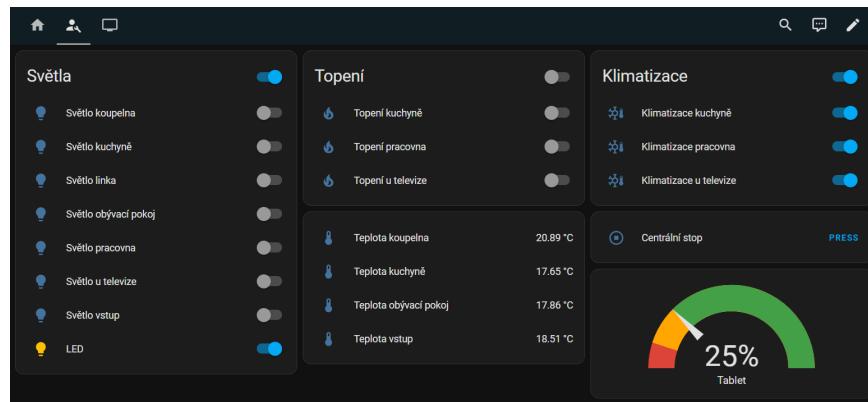
Pro komunikaci mezi Home Assistantem a PLC byla použita integrace *MQTT*, která se nastavila pomocí webového rozhraní a konfiguračního souboru YAML (Příloha H). Webové rozhraní slouží k nastavení IP adresy, portu, uživatelského jména a hesla pro připojení k brokeru. YAML soubor slouží pro vytvoření jednotlivých entit.

Po vytvoření všech entit jim byl přidělen název, ikona a byly rozděleny do různých skupin podle místa použití (záložka *Area*). Důsledkem tohoto kroku byla automaticky vygenerována vizualizace, která sloužila k hrubému ovládání a sledování stavu jednotlivých zařízení (Obr. 4.4).

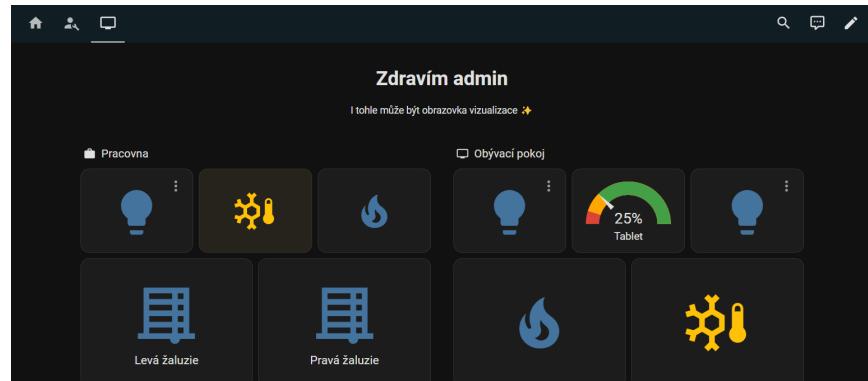


Obr. 4.4: Home Assistant hrubá vizualizace

Pro jednodušší ovládání a přehlednost byly vytvořeny vlastní vizualizace. První byla vytvořena podle skupin ovládaných prvků (Obr. 4.5), druhá byla vytvořena podle místností (Obr. 4.6). Dále na nich byly použity jiné možné vzhledy jednotlivých prvků, které nabízí základní verze Home Assistantu. Pro případ, že by se ani ty uživateli nelíbily, je možné otevřít vyskakovací okna jednotlivých prvků, které nabízí více možností ovládání. V těchto oknech lze také sledovat historii jednotlivých prvků.



Obr. 4.5: Home Assistant funkční vizualizace



Obr. 4.6: Home Assistant vizualizace dle umístění

Pro potřebu sledování historie jednotlivých zařízení je možné použít vestavěnou funkci Home Assistantu - *History*, která zobrazuje blízkou historii. Pro sledování delší historie bohužel neexistuje adekvátní řešení, protože Home Assistant ukládá historii do SQLite databáze, která není určena pro dlouhodobé ukládání dat. Proto byla použita integrace *InfluxDB*. Další výhodou použití externí databáze je možnost sledovat historii i bez připojení k Home Assistantu.

Připojení této databáze je uvedeno v příloze H.

4.5 InfluxDB

Jedná se o open-source databázi určenou pro ukládání časových řad. Databáze je optimalizována pro čtení a zápis velkého množství dat, která se následně využívají k analýze v reálném čase. Její hlavní výhodou je rychlosť a efektivita při práci s velkým množstvím dat. Pro udržení rychlosti byly implementovány retenční politiky, které určují, jak dlouho se data uchovávají a jakým způsobem se agregují. [51]

Databáze je tvořena několika základními koncepty [52]:

- **Bucket** – kontejner pro ukládání dat
- **Measurement** – popis dat ukládaných do bucketu
- **Series** – skupina bodů se stejným tagem a měřením
- **Point** – jednotkový záznam v bucketu
- **Field** – hodnota ukládaná do bucketu
- **Tag** – klíč používaný k identifikaci dat v bucketu
- **Timestamp** – časový údaj o uložení dat
- **Retention policy** – pravidlo pro uchovávání dat v bucketu
- **Continuous query** – dotaz, který se provádí automaticky v pravidelných intervalech.

Kontejner této databáze běží na portu 8086. Po vytvoření kontejneru je potřeba nastavit uživatelské jméno a heslo pro přístup do databáze. Dále je potřeba nastavit bucket, do kterého se budou ukládat data – základní informace, délku uchovávání, způsob agregace dat a přístup přes API. Všechny tyto informace se nastavují pomocí webového rozhraní.

Webové rozhraní také umožňuje vytvářet a spravovat uživatelské účty a přístupová práva, sledovat výkon databáze, historii jednotlivých dat a vytvářet pro ně dotazy, které se provádějí pomocí jazyka *Flux*. Tyto dotazy slouží k analýze dat a vytváření grafů, a to buď přímo v InfluxDB, nebo pomocí externích nástrojů, jako je Grafana.

4.6 Grafana

Grafana je open-source platforma pro vizualizaci a analýzu dat. Umožňuje uživatelům vytvářet interaktivní grafy, tabulky a panely pro zpracování dat z různých zdrojů a také nastavovat upozornění na základě těchto dat. Disponuje širokým portfoliem pluginů, které umožňují připojení k různým databázím a službám. Grafana je velmi populární mezi vývojáři a administrátory, kteří potřebují monitorovat a analyzovat data v reálném čase. [53]

Vizualizace v Grafaně je tvořena pomocí přehledů (dashboardů), které obsahují panely. Tyto panely mohou mít různé typy grafů [54]:

- **Grafy a diagramy**
 - **Time series** – data v čase
 - **State timeline** – stav v čase
 - **Status history** – periodický stav v čase
 - **Bar chart** – kategoriální data
 - **Histogram** – rozložení dat prezentované jako sloupcový graf
 - **Heatmap** – vizualizace dat ve dvou rozměrech, typicky velikosti jevu
 - **Pie chart** – zobrazení proporcionality
 - **Candlestick** – typicky pro finanční data
 - **Gauge** – zaoblený ukazatel zobrazující, jak daleko je metrika od prahu
 - **Trend** – datové sady se sekvenčním, číselným x, které není čas
 - **XY chart** – vizualizace hodnot x a y v grafu
- **Statistika a čísla**
 - **Stat** – velké statistiky a změny v čase
 - **Bar gauge** – horizontální nebo vertikální ukazatel
- **Různé**
 - **Table** – vizualizace dat v tabulce
 - **Logs** – vizualizace pro protokoly
 - **Node graph** – orientované grafy nebo síť
 - **Traces** – trasování
 - **Flame graph** – profilování
 - **Canvas** – explicitní umístění prvků do statických a dynamických rozvržení
 - **Geomap** – geoprostorová data
 - **Datagrid** – vytváření a manipulace s daty (slouží jako zdroj dat pro jiné panely)
- **Widgety**
 - **Dashboard list** – seznam přehledů
 - **Alert list** – seznam upozornění
 - **Annotations list** – seznam anotací
 - **Text** – markdown a HTML
 - **News** – RSS kanály

Stejně jako v předchozích podkapitolách je i zde možné vytvářet a spravovat uživatelské účty a přístupová práva prostřednictvím webového rozhraní, tentokrát na portu 3000.

Grafana v této práci sloužila jako nástroj pro zpracování historických dat instalace, konkrétně pro sledování spotřeby energie, jejího rozložení, měsíčních nákladů a historie sepnutí jednotlivých zařízení. Implementace a vzhled jednotlivých grafů jsou uvedeny v příloze I.

Závěr

V rámci této bakalářské práce byly splněny všechny stanovené cíle týkající se návrhu a realizace vzdáleného řízení a vizualizace demonstračního panelu KNX pro ovládání funkcí osvětlení, žaluzií, topení a klimatizace. Práce poskytuje ucelený pohled na problematiku sběrnicového systému KNX, jeho možnosti a praktické využití v oblasti domácí automatizace.

V první části práce byla provedena důkladná analýza technologie KNX, včetně její historie, základních principů fungování, struktury komunikace, zabezpečení a topologie. Tato část poskytla nezbytný teoretický základ pro následnou praktickou realizaci.

Druhá část práce se zaměřila na parametrizaci konkrétních přístrojů a tvorbu dynamických světelných scén. Byly popsány jednotlivé kroky od návrhu projektu až po jeho implementaci, včetně identifikace problémů.

Třetí část práce se věnovala samotnému řízení a vizualizaci prostřednictvím PLC společnosti TECO. Popsány byly nejen technické aspekty programování a komunikace mezi PLC a KNX, ale také implementace MQTT protokolu a tvorba webové vizualizace. V této části práce vznikl problém těsně před jejím dokončením – došlo k poruše komunikační brány KNX IP BAOS 774. To mělo za následek nefunkčnost komunikace mezi PLC a KNX instalací. Situace je nyní řešena s technickou podporou a v nejhorším případě bude nutné vyměnit celou bránu.

V poslední části byly zhodnoceny možnosti využití open-source platform pro vizualizaci a správu domácí automatizace. Byla provedena implementace na jednodeskovém počítači Raspberry Pi s využitím kontejnerizace Docker a nasazením několika open-source nástrojů, jako jsou Home Assistant, InfluxDB či Grafana. Tyto nástroje byly vybrány a nasazeny s ohledem na jejich aktuální popularitu, dostupnost a možnosti rozšíření.

Celkově lze konstatovat, že práce splnila vytýčené cíle a přinesla komplexní řešení vzdáleného řízení a vizualizace KNX panelu. Byly ověřeny možnosti integrace různých technologií a platform, přičemž důraz byl kladen na praktičnost, flexibilitu a budoucí rozšiřitelnost řešení. Výsledky práce mohou sloužit jako inspirace či základ pro další rozvoj v oblasti chytré domácnosti a automatizace budov.

Literatura

- [1] Asociace KNX *Datapoint Type*. Online. Dostupné z: <https://support.knx.org/hc/en-us/articles/115001133744-Datapoint-Type> [cit. 23. 12. 2021].
- [2] Asociace KNX *A History of KNX*. Online. Dostupné z: https://crelectrics.com.au/wp-content/uploads/2015/05/a_history_of_KNX.pdf [cit. 1. 10. 2021].
- [3] Asociace KNX *KNX Basics*. Online. Dostupné z: https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics_cz.pdf [cit. 1. 10. 2021].
- [4] Asociace KNX *Principy systému KNX*. Online. Dostupné z: https://knxcz.cz/images/clanky/KNX-System-Principles_cz.pdf [cit. 1. 10. 2021].
- [5] Asociace KNX *KNX Secure Devices*. Online. Dostupné z: <https://support.knx.org/hc/en-us/articles/360000216419-KNX-Secure-Devices> [cit. 23. 12. 2021].
- [6] Asociace KNX *ISO/IEC 14543-3. KNX Celkový přehled*.
- [7] Asociace KNX *ISO/IEC 14543-3. KNX Systémové argumenty*.
- [8] Asociace KNX *ISO/IEC 14543-3. KNX TP Topologie*.
- [9] MITRENGA, Michal.: *Realizace demonstrativního panelu inteligentní elektroinstalace KNX*. Brno, 2021.. Online. [cit. 26. 12. 2021]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134788> Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Fiedler.
- [10] knx.org. Online. Dostupné z: <https://www.knx.org> [cit. 1. 10. 2021].
- [11] knx.org *ETS Professional*. Online. Dostupné z: <https://www.knx.org/knx-en/for-professionals/software/ets-professional/> [cit. 2. 1. 2022].
- [12] Asociace KNX *System Specifications KNXnet/IP - Tunelling*. Online. Dostupné z: <https://community-openhab-org.s3-eu-central-1.amazonaws.com/original/2X/8/8b3ec554f60872e37763d2005edc1c4c1fb16887.PDF> [cit. 5. 5. 2025].
- [13] Asociace KNX *System Specifications - KNXnet/IP - Routing*. Online. Dostupné z: <https://community-openhab-org.s3-eu-central-1.amazonaws.com/>

com/original/2X/b/ba93de8a703a5ece40f0dfc1b596643cb28e8497.PDF
[cit. 5. 5. 2025].

- [14] ABB - SBR/U6.0.1-84. Online. Dostupné z: <https://new.abb.com/products/2CKA006330A0004/sbr-u6-0-1-84> [cit. 28. 12. 2021].
- [15] ABB - SA/S8.10.2.1. Online. Dostupné z: <https://new.abb.com/products/2CDG110157R0011/sa-s8-10-2-1> [cit. 28. 12. 2021].
- [16] ABB - JRA/S4.230.2.1. Online. Dostupné z: <https://new.abb.com/products/2CDG110121R0011/jra-s4-230-2-1> [cit. 28. 12. 2021].
- [17] Basalte - Sentido aluminium - quad - Brushed black. Online. Dostupné z: <https://www.knxstore.cz/domu/1000403-basalte-sentido-aluminium-quad-brushed-black-5425025030224.html> [cit. 28. 12. 2021].
- [18] B.E.G - Indoor 140-L-KNX-DX. Online. Dostupné z: <https://www.beg-luxomat.com/cz/produkty/luxomatnet/knx/knx-gen6-deluxe-pritomnostni-detektor/indoor-140-l-knx-dx/> [cit. 28. 12. 2021].
- [19] Berker - B.IQ push-button 3gang with thermostat Display, KNX. Online. Dostupné z: <https://www.berker.com/en/e-catalogue/building-management-systems/knx-systems/berker-knx-system/b.iq-push-buttons-with-thermostat/75663593/355802.htm?lang=en> [cit. 28. 12. 2021].
- [20] EKINEX - Pushbutton with thermostat. Online. Dostupné z: <https://www.ekinex.com/en/15/pushbutton-with-thermostat.html> [cit. 28. 12. 2021].
- [21] HDL - M/TBP6.1-A2-46 Ovládací prvek 6násobný iTouch, bílá. Online. Dostupné z: <https://b2b.hdl-automation.cz/cz/produkty/knx/ovladaci-prvky-hdl/ovladaci-prvky-itouch/hdl-m-tbp6-1-a2-46> [cit. 28. 12. 2021].
- [22] HDL - M/R8.10.1 8CH 10A High Power Switch Actuator. Online. Dostupné z: <https://b2b.hdl-automation.cz/en/products/knx/switching-actuators/hdl-m-r8-10-1> [cit. 28. 12. 2021].
- [23] HDL - M/DRGW4.1 Akční člen stmívací LED 4násobný, 7 A. Online. Dostupné z: <https://b2b.hdl-automation.cz/cz/produkty/knx/akcni-cleny-stmivaci/hdl-m-drgbw4-1> [cit. 28. 12. 2021].

- [24] SIEMENS - QMX3.P37 *Prostorový KNX přístroj, displej pro regulaci HVAC, čidlo teploty, konfigurovatelná tlačítka pro osvětlení/žaluzie/scény*. Online. Dostupné z: <https://hit.sbt.siemens.com/RWD/app.aspx?RC=CZ&lang=cs&MODULE=Catalog&ACTION>ShowProduct&KEY=S55624-H108> [cit. 28. 12. 2021].
- [25] SIEMENS - 5WG1 146-1AB03. Online. Dostupné z: https://www.hqs.sbt.siemens.com/cps_product_data/data/search_find_en.htm?ssn=5WG11461AB03 [cit. 28. 12. 2021].
- [26] Simon *Standard button box 4 functions white Simon 82 Sense*. Online. Dostupné z: <https://www.simonelectric.com/intl/8000641-030-standard-button-box-4-functions-white-simon-82-sense.html> [cit. 28. 12. 2021].
- [27] Weinzierl - KNX IP BAOS 774. Online. Dostupné z: <https://www.weinzierl.de/index.php/en/all-knx/knx-devices-en/knx-ip-baos-774-en> [cit. 28. 12. 2021].
- [28] Weinzierl - KNX IP BAOS 774 *Rozhraní BAOS do 1000 bodů*. Online. Dostupné z: <https://knx-trade.ru/weinzierl/597-5263.html> [cit. 28. 12. 2021].
- [29] TECO *CP-2007*. Online. Dostupné z: <https://wiki.tecomat.cz/clanek/cp-2007> [cit. 23. 4. 2025].
- [30] TECO *Mosaic*. Online. Dostupné z: <https://www.tecomat.cz/ke-stazeni/software/mosaic/> [cit. 23. 4. 2025].
- [31] TECO *Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic*. Online. Dostupné z: <https://catalog.tecomat.cz/produkt/programovani-dle-normy-iec-61-131#download> [cit. 23. 4. 2025].
- [32] TECO *Knihovna KnxLib*. Online. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00380_01 [cit. 23. 4. 2025].
- [33] MQTT *The Standard for IoT Messaging*. Online. Dostupné z: <https://mqtt.org> [cit. 15. 4. 2025].
- [34] MQTT Essentials *The Ultimate Guide to MQTT for Beginners and Experts*. Online. Dostupné z: <https://www.hivemq.com/mqtt/>
- [35] JSON *JavaScript Object Notation*. Online. Dostupné z: <https://www.json.org/json-en.html> [cit. 15. 4. 2025].

- [36] TECO *Knihovna MQTTLib*. Online. Dostupné z: <https://support.tecomat.cz/storage/app/uploads/public/633/acd/862/633acd8625f3d859405244.pdf> [cit. 23. 4. 2025].
- [37] TECO *WebMaker*. Online. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00328_01_mosaic_webmaker_cz [cit. 23. 4. 2025].
- [38] Flaticon *The most wanted free SVG user interface icons*. Online. Dostupné z: <https://www.flaticon.com/> [cit. 5. 5. 2025].
- [39] Raspberry Pi *Raspberry Pi 5*. Online. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-5/> [cit. 5. 5. 2025].
- [40] KIOXIA *Managed Flash Background Operations Series - Part 3: Understanding Wear Leveling in NAND Flash Memory*. Online. Dostupné z: https://americas.kioxia.com/content/dam/kioxia/en-us/business/memory/mlc-nand/asset/KIOXIA_Managed_Flash_BOS_P3_Understanding_Wear_Leveling_Tech_Brief.pdf [cit. 5. 5. 2025].
- [41] Docker Compose. Online. Dostupné z: <https://docs.docker.com/compose/> [cit. 5. 5. 2025].
- [42] Docker *Docker Installation for Debian*. Online. Dostupné z: <https://docs.docker.com/engine/install/debian/> [cit. 5. 5. 2025].
- [43] Linux Containers *Container and Virtualization tools*. Online. Dostupné z: <https://linuxcontainers.org> [cit. 5. 5. 2025].
- [44] Yaml *YAML Ain't Markup Language™*. Online. Dostupné z: <https://yaml.org/> [cit. 5. 5. 2025].
- [45] Docker *Compose file version 3 reference*. Online. Dostupné z: <https://docs.docker.com/compose/compose-file/> [cit. 5. 5. 2025].
- [46] Docker *Environment variables*. Online. Dostupné z: <https://docs.docker.com/engine/reference/run/#env> [cit. 5. 5. 2025].
- [47] Portainer. Online. Dostupné z: <https://www.portainer.io/> [cit. 5. 5. 2025].
- [48] Eclipse Mosquitto. Online. Dostupné z: <https://mosquitto.org/> [cit. 15. 5. 2025].
- [49] Home Assistant. Online. Dostupné z: <https://www.home-assistant.io/> [cit. 15. 5. 2025].

- [50] Home Assistant Community Store. Online. Dostupné z: <https://hacs.xyz/> [cit. 15. 5. 2025].
- [51] InfluxDB. Online. Dostupné z: <https://www.influxdata.com/> [cit. 15. 5. 2025].
- [52] InfluxDB *Key Concepts*. Online. Dostupné z: https://docs.influxdata.com/influxdb/v1/concepts/key_concepts/ [cit. 15. 5. 2025].
- [53] Grafana. Online. Dostupné z: <https://grafana.com/> [cit. 15. 5. 2025].
- [54] Grafana *Visualizations*. Online. Dostupné z: <https://grafana.com/docs/grafana/latest/panels-visualizations/visualizations/> [cit. 15. 5. 2025].

Seznam symbolů a zkratek

API	Application Programming Interface (<i>Programové rozhraní aplikace</i>)
BCI	Batibus Club International (<i>Mezinárodní spolek Batibus</i>)
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance (<i>Vícenásobný přístup s detekcí obsazenosti nosiče a vyhýbání se kolizím</i>)
DPT	Datapoint (<i>Datový bod</i>)
EIBA	European Installation Bus Association (<i>Evropská asociace instalacích sběrnic</i>)
EHSA	European Home Systems Association (<i>Evropská asociace domácích systémů</i>)
ETS	Engineering Tool Software (<i>Software pro inženýrské nástroje</i>)
FDSK	Factory Default Setup Key (<i>Výchozí klíč továrního nastavení</i>)
IA	Individual Address (<i>Individuální adresa</i>)
ICT	Information and Communication Technology (<i>Informační a komunikační technologie</i>)
IoT	Internet of Things (<i>Internet věcí</i>)
IP	Internetový protokol
ISDN	Digitální síť integrovaných služeb
LED	Elektroluminiscenční dioda
LS	Liniová spojka
MAC	Message Authentication Code (<i>kód pro ověřování zpráv</i>)
MQTT	Message Queuing Telemetry Transport (<i>Protokol přenosu telemetrie s frontou zpráv</i>)
NO	Normally Opened (<i>Normálně sepnuté</i>)
NZ/TI	Zdroj s tlumivkou
OS	Oblastní spojka
PL	Powerline (<i>Přenos po elektrické síti</i>)

PLC	Programovatelný logický automat
QoS	Quality of Service (<i>Kvalita služby</i>)
RF	Radiofrekvenční komunikační médium
RGB	Red–Green–Blue (<i>červená–zelená–modrá</i>)
RSS	Really Simple Syndication (<i>Opravdu jednoduché syndikování</i>)
SSL	Secure Sockets Layer (<i>Zabezpečení soketové vrstvy</i>)
TLS	Transport Layer Security (<i>Zabezpečení přenosové vrstvy</i>)
TP	Twisted Pair (<i>Kroucená dvojlinka</i>)
US	Účastník sběrnice
WLAN	Wireless Local Area Network (<i>Bezdrátová místní síť</i>)

Seznam příloh

A Skupinové adresy	95
B Globální proměnné a struktury PLC	101
C Definice funkčních složitějších funkčních bloků	103
C.0.1 fbRoomTempMod	103
C.0.2 Popis funkčního bloku fbBath	108
C.0.3 fbKitch	109
C.0.4 fbLivRoom	111
C.0.5 fbOutz	114
D Program komunikace mezi PLC a KNX	115
E Program komunikace mezi PLC a Home Assistantem	119
F Vizualizace pomocí WebMakeru	125
G Docker YAML soubory	129
G.0.1 Portainer YAML	129
G.0.2 Stack YAML	130
H Home Assistant - konfigurační soubor	133
I Nastavení přehledů v Grafaně	135
I.0.1 Grafana - Spotřeba	135
I.0.2 Grafana - Spínání	138
I.0.3 Grafana - Celkové měsíční náklady	140
I.0.4 Grafana - Průběh teploty	142

A Skupinové adresy



Skupinová adresa

Projekt: Kelnar SP

Počáteční datum: pondělí 29. listopadu 2021

Datum importu: pondělí 29. listopadu 2021

Tisknout datum: **pondělí 10. ledna 2022**

Čas tisku: 10:47:57

Stav: Editace

Adresa	Název	Délka	Centrál	Vést přes liniovou spojku
Popis				
Komentáře				
0	Světla			Ne
0/0	Jednotlive			Ne
0/0/1	S1_kuch	switch	Ne	Ne
0/0/2	S2_kuch_linka	switch	Ne	Ne
0/0/3	S3_prac	switch	Ne	Ne
0/0/4	S4_obyv	switch	Ne	Ne
0/0/5	S5_TV	switch	Ne	Ne
0/0/6	S6_koup	switch	Ne	Ne
0/0/7	S7_vstup	switch	Ne	Ne
0/1	Skupiny			Ne
0/1/0	S1,S2,S6_prizemi	switch	Ne	Ne
0/1/1	S3,S4,S5_patro	switch	Ne	Ne
0/2	Status			Ne
0/2/0	S1_Status_kuch	switch	Ne	Ne
0/2/1	S2_Status_kuch_linka	switch	Ne	Ne
0/2/2	S3_Status_prac	switch	Ne	Ne
0/2/3	S4_Status_obyv	switch	Ne	Ne
0/2/4	S5_Status_TV	switch	Ne	Ne
0/2/5	S6_Status_koup	switch	Ne	Ne
0/2/6	S7_Status_vstup	switch	Ne	Ne
1	Žaluzie			Ne
1/0	Pohyb			Ne
1/0/0	Ž1_pohyb	up/down	Ne	Ne
1/0/1	Ž2_pohyb	up/down	Ne	Ne
1/0/2	Ž1,Ž2_pohyb	switch	Ne	Ne
1/1	Krok			Ne
1/1/0	Ž1_krok	switch	Ne	Ne
1/1/1	Ž2_krok	1-bit	Ne	Ne
1/1/2	Ž1,Ž2_krok	step	Ne	Ne
2	Topení			Ne
2/0	Jednotlive			Ne
2/0/0	T1_prac	switch	Ne	Ne
2/0/1	T2_obyv	switch	Ne	Ne
2/0/2	T3_kuch	switch	Ne	Ne
2/0/3	T3/1_kuch	switch	Ne	Ne
2/0/4	T3/2_kuch	switch	Ne	Ne
2/1	Skupiny			Ne
2/1/3	T1,T2,T3	switch	Ne	Ne
2/2	Status			Ne
2/2/0	T1/1_prac_STATUS	switch	Ne	Ne
2/2/1	T1/2_prac_STATUS	switch	Ne	Ne
2/2/2	T2/1_obyv_STATUS	switch	Ne	Ne

Adresa	Název	Délka	Centrál	Vést přes liniovou spojku
Popis	Komentáře			
2/2	Status			Ne
2/2/3	T2/2_obyv_STATUS	switch	Ne	Ne
2/2/4	T3/1_kuch_STATUS	switch	Ne	Ne
2/2/5	T3/2_kuch_STATUS	switch	Ne	Ne
3	Klimatizace			Ne
3/0	Jednotlivé			Ne
3/0/0	Ch1_prac	switch	Ne	Ne
3/0/1	Ch2_TV	switch	Ne	Ne
3/0/2	Ch3_kuch	switch	Ne	Ne
3/1	Skupiny			Ne
3/1/0	Ch1,Ch2,Ch3	switch	Ne	Ne
3/1/1	Ch1,Ch2	switch	Ne	Ne
3/2	Status			Ne
3/2/0	Ch1_obyv_leva_STATUS	switch	Ne	Ne
3/2/1	Ch2_obyv_prava_STATUS	switch	Ne	Ne
3/2/2	Ch3_kuch_STATUS	switch	Ne	Ne
4	LED			Ne
4/0	Zapnutí			Ne
4/0/0	RGBW_Red	switch	Ne	Ne
4/0/1	RGBW_Green	switch	Ne	Ne
4/0/2	RGBW_Blue	switch	Ne	Ne
4/0/3	RGBW_White	switch	Ne	Ne
4/1	Stmívání			Ne
4/1/0	RGBW_St_Red	dimming control	Ne	Ne
4/1/1	RGBW_St_Green	dimming control	Ne	Ne
4/1/2	RGBW_St_Blue	dimming control	Ne	Ne
4/1/3	RGBW_St_White	dimming control	Ne	Ne
4/2	RGB sety			Ne
4/2/0	RGBW_Set1	3 bytes	Ne	Ne
4/2/1	RGBW_Set2	3 bytes	Ne	Ne
4/2/2	RGBW_Set3	3 bytes	Ne	Ne
4/2/3	RGBW_Set4	3 bytes	Ne	Ne
4/2/4	RGBW_Set5	3 bytes	Ne	Ne
4/3	Status			Ne
4/3/0	S_Blue_STATUS	switch	Ne	Ne
4/3/1	S_Green_STATUS	switch	Ne	Ne
4/3/2	S_Red_STATUS	switch	Ne	Ne
4/3/3	S_White_STATUS	switch	Ne	Ne
4/3/4	RGB_STATUS color	3 bytes	Ne	Ne
4/3/5	RGB_STATUS	switch	Ne	Ne
5	Scény			Ne
5/0	Světla			Ne

Adresa	Název	Délka	Centrál	Vést přes liniovou spojku
Popis				
Komentáře				
5/0	Světla			Ne
5/0/0	SC_S1,S2,S3,S4,S5,S6	switch	Ne	Ne
5/1	Žaluzie			Ne
5/1/0	SC_Ž1,Ž2_P	switch	Ne	Ne
5/2	Topení			Ne
5/2/2	SC_T1,T2,T3	switch	Ne	Ne
5/3	Klimatizace			Ne
5/3/0	SC_Ch1,Ch2,Ch3	switch	Ne	Ne
5/4	LED			Ne
5/4/0	SC_LED_Red	switch	Ne	Ne
5/4/1	SC_LED_Green	switch	Ne	Ne
5/4/2	SC_LED_Blue	switch	Ne	Ne
5/4/3	SC_LED_White	switch	Ne	Ne
5/4/4	SC_LED_Yellow	3 bytes	Ne	Ne
5/4/5	SC_LED_Vypnuto	3 bytes	Ne	Ne
5/5	Volání			Ne
5/5/0	SC_Příchod	scene control	Ne	Ne
5/5/1	SC_Odchod	1 byte	Ne	Ne
5/5/2	SC_Dovolená	1 byte	Ne	Ne
5/5/3	SC_Léto	scene control	Ne	Ne
5/5/4	SC_Zima	scene control	Ne	Ne
6	Měření			Ne
6/0	Teplota			Ne
6/0/0	Temp_Siemens	temperature (°C)	Ne	Ne
6/0/1	Temp_ABB	temperature (°C)	Ne	Ne
6/0/2	Temp_Ekinex	temperature (°C)	Ne	Ne
6/0/3	Temp_BEG	temperature (°C)	Ne	Ne
6/0/4	Temp_Basalte	temperature (°C)	Ne	Ne
6/0/5	Temp_Simon	temperature (°C)	Ne	Ne

B Globální proměnné a struktury PLC

Výpis B.1: Globální proměnné a struktury PLC

```
1 TYPE
2     DT_CMD_BOOL : STRUCT
3         CMD_VAL : BOOL;
4         CMD : BOOL;
5     END_STRUCT;
6 END_TYPE
7 VAR_GLOBAL
8     SV1_FB      : BOOL := FALSE; //KNX COMM SV1 FB
9     SV2_FB      : BOOL := FALSE; //KNX COMM SV2 FB
10    SV3_FB      : BOOL := FALSE; //KNX COMM SV3 FB
11    SV4_FB      : BOOL := FALSE; //KNX COMM SV4 FB
12    SV5_FB      : BOOL := FALSE; //KNX COMM SV5 FB
13    SV6_FB      : BOOL := FALSE; //KNX COMM SV6 FB
14    SV7_FB      : BOOL := FALSE; //KNX COMM SV6 FB
15    SV1_CMD     : DT_CMD_BOOL; //KNX CMD SV1
16    SV2_CMD     : DT_CMD_BOOL; //KNX CMD SV2
17    SV3_CMD     : DT_CMD_BOOL; //KNX CMD SV3
18    SV4_CMD     : DT_CMD_BOOL; //KNX CMD SV4
19    SV5_CMD     : DT_CMD_BOOL; //KNX CMD SV5
20    SV6_CMD     : DT_CMD_BOOL; //KNX CMD SV6
21    SV7_CMD     : DT_CMD_BOOL; //KNX CMD SV6
22    HEAT1_FB    : BOOL := FALSE; //KNX COMM HEAT1 FB
23    HEAT2_FB    : BOOL := FALSE; //KNX COMM HEAT2 FB
24    HEAT3_FB    : BOOL := FALSE; //KNX COMM HEAT3 FB
25    HEAT1_CMD   : DT_CMD_BOOL; //KNX CMD HEAT1
26    HEAT2_CMD   : DT_CMD_BOOL; //KNX CMD HEAT2
27    HEAT3_CMD   : DT_CMD_BOOL; //KNX CMD HEAT3
28    COLD1_FB    : BOOL := FALSE; //KNX COMM COLD1 FB
29    COLD2_FB    : BOOL := FALSE; //KNX COMM COLD2 FB
30    COLD3_FB    : BOOL := FALSE; //KNX COMM COLD3 FB
31    COLD1_CMD   : DT_CMD_BOOL; //KNX CMD COLD1
32    COLD2_CMD   : DT_CMD_BOOL; //KNX CMD COLD2
33    COLD3_CMD   : DT_CMD_BOOL; //KNX CMD COLD3
34    SHUT1_FB_UP : BOOL := FALSE; //KNX COMM SHUT1 FB
35    SHUT1_FB_DOWN : BOOL := FALSE; //KNX COMM SHUT1 FB
36    SHUT1_CMD   : DT_CMD_BOOL; //KNX CMD SHUT1
37    SHUT1_STEP_CMD : DT_CMD_BOOL; //KNX CMD SHUT1 STEP
```

```

38 SHUT2_FB_UP      : BOOL := FALSE; //KNX COMM SHUT2 FB
39 SHUT2_FB_DOWN    : BOOL := FALSE; //KNX COMM SHUT2 FB
40 SHUT2_CMD        : DT_CMD_BOOL; //KNX CMD SHUT1
41 SHUT2_STEP_CMD   : DT_CMD_BOOL; //KNX CMD SHUT1 STEP
42 KNX_TEMPER       : REAL := 0.0; //KNX TEMPERATURE OUT
43 OUTSIDE_TEMPER   : REAL := 0.0; //OUTSIDE TEMPER FOR MQTT
44 KITCHEN_TEMPER   : REAL := 0.0; //KITCHEN TEMPER FOR MQTT
45 LIVINGR_TEMPER   : REAL := 0.0; //LIVINGROOM TEMPER FOR MQTT
46 BATHROOM_TEMPER  : REAL := 0.0; //BATHROOM TEMPER FOR MQTT
47 SHUTDOWN_MQTT    : BOOL := FALSE; //SHUTDOWN FROM MQTT
48 END_VAR

```

C Definice funkčních složitějších funkčních bloků

C.0.1 fbRoomTempMod

Výpis C.1: fbRoomTempMod

```
1 FUNCTION_BLOCK fbRoomTempMod
2   VAR_INPUT
3     Heat_1          : BOOL; // Topení vstup 1[-]
4     Heat_1_WATTS   : REAL; // Topení výkon 1[W] =>[J/s]
5     Heat_2          : BOOL; // Topení vstup 2[-]
6     Heat_2_WATTS   : REAL; // Topení výkon 2[W] =>[J/s]
7     Cold_1          : BOOL; // Klimatizace vstup 1[-]
8     Cold_1_WATTS   : REAL; // Klimatizace výkon 1[W] =>[J/s]
9     Cold_2          : BOOL; // Klimatizace vstup 2[-]
10    Cold_2_WATTS   : REAL; // Klimatizace výkon 2[W] =>[J/s]
11    lenght         : REAL; // délka[m]
12    width          : REAL; // šířka[m]
13    height         : REAL; // výška[m]
14    wall_temp1     : REAL; // Teplota za sousední zdí[degC]
15    wall_temp2     : REAL; // Teplota za sousední zdí[degC]
16    wall_temp3     : REAL; // Teplota za sousední zdí[degC]
17    wall_temp4     : REAL; // Teplota za sousední zdí[degC]
18    floor_temp     : REAL; // Teplota v místnosti pod[degC]
19    ceiling_temp   : REAL; // Teplota v místnosti nad[degC]
20    wall_thic1     : REAL; // Šířka zdi1[m]
21    wall_thic2     : REAL; // Šířka zdi2[m]
22    wall_thic3     : REAL; // Šířka zdi3[m]
23    wall_thic4     : REAL; // Šířka zdi4[m]
24    floor_thic    : REAL; // Šířka podlahy[m]
25    ceiling_thic  : REAL; // Šířka stropu[m]
26    TaskTime       : REAL; // Rychlosť tasku[ms]
27  END_VAR
28  VAR_OUTPUT
29    Temperature    : REAL := 20.0; // Teplota na výstupu[degC]
30  END_VAR
31  VAR
32    INIT           : BOOL := FALSE; //INIT bloku
33    TimeStep       : REAL := 0.0; // Hodnota kroku v ms
```

```

34    VAir      : REAL := 0.0; // Obsah vzduchu[m^3]
35    MAir      : REAL := 0.0; // Váha vzduchu[Kg]
36    QAir      : REAL := 0.0; // Energie změna o 1degC[J]
37    RoomTemp   : REAL := 20.0; // Pokojová teplota[degC]
38    DeltaTemp   : REAL := 0.0; // Teplota za cyklus[degC]
39    KHeatRise   : REAL := 2.4; // Korekční člen tr[-]
40    KColdRise   : REAL := 45.0; // Korekční člen kr[-]
41    KHeatFall1  : REAL := 0.0; // Korekční člen tf 1[-]
42    KColdFall1  : REAL := 0.0; // Korekční člen kf 1[-]
43    KHeatFall2  : REAL := 0.0; // Korekční člen tf 2[-]
44    KColdFall2  : REAL := 0.0; // Korekční člen kf 2[-]
45    Epsilon     : REAL := 1.0; // Sp výkon za čas[w]
46    AlphaHeat    : REAL := 0.0; // Logaritmus topení[-]
47    AlphaCold    : REAL := 0.0; // Logaritmus klimatizace[-]
48    FiTotal     : REAL := 0.0; // Celkový tepelný tok[J/s]
49    FiHeat       : REAL := 0.0; // Celkový tepelný tok t[J/s]
50    FiHeatTmp1   : REAL := 0.0; // Tepelný tok topení 1[J/s]
51    FiHeatTmp2   : REAL := 0.0; // Tepelný tok topení 2[J/s]
52    FiCold       : REAL := 0.0; // Celkový tepelný tok k[J/s]
53    FiColdTmp1   : REAL := 0.0; // Tepelný tok klim 1[J/s]
54    FiColdTmp2   : REAL := 0.0; // Tepelný tok klim 2[J/s]
55    AreaWall1   : REAL := 0.0; // Plocha zdi 1[m^2]
56    AreaWall2   : REAL := 0.0; // Plocha zdi 2[m^2]
57    AreaWall3   : REAL := 0.0; // Plocha zdi 3[m^2]
58    AreaWall4   : REAL := 0.0; // Plocha zdi 4[m^2]
59    AreaFloor    : REAL := 0.0; // Plocha podlahy[m^2]
60    AreaCeiling  : REAL := 0.0; // Plocha stropu[m^2]
61 END_VAR
62 VAR_TEMP
63    DeltaTempWall_1 : REAL := 0; // Rozdíl 1[degC]
64    DeltaTempWall_2 : REAL := 0; // Rozdíl 2[degC]
65    DeltaTempWall_3 : REAL := 0; // Rozdíl 3[degC]
66    DeltaTempWall_4 : REAL := 0; // Rozdíl 4[degC]
67    DeltaTempFloor  : REAL := 0; // Rozdíl podlaha[degC]
68    DeltaTempCeiling: REAL := 0; // Rozdíl strop[degC]
69    FiWall_1       : REAL := 0.0; // Tepelný tok 1[J/s]
70    FiWall_2       : REAL := 0.0; // Tepelný tok 2[J/s]
71    FiWall_3       : REAL := 0.0; // Tepelný tok 3[J/s]
72    FiWall_4       : REAL := 0.0; // Tepelný tok 4[J/s]
73    FiFloor        : REAL := 0.0; // Tepelný tok podlaha[J/s]

```

```

74     FiCeiling      : REAL := 0.0; // Tepelný tok strop [J/s]
75 END_VAR
76 VAR CONSTANT
77     RoAir : REAL := 1.204; // Hustota vzduchu [Kg/m^3]
78     CpAir : REAL := 1005.0; // Tepelná kap vzduchu [J/(kg*K)]
79     LambdaBrick : REAL := 0.4; // Tepelná vod cihly [W/(m*K)]
80     MaxTemp      : REAL := 24.0; // Maximální teplota [degC]
81     MinTemp      : REAL := 16.0; // Minimální teplota [degC]
82     TimeRise     : REAL := 15.0; // Čas náběhu výkonu [s]
83     TimeFallHeat : REAL := 15.0; // Čas klesání výkonu t [s]
84     TimeFallCold : REAL := 10.0; // Čas klesání výkonu k [s]
85     TargTimeHeat : REAL := 5.06; // Čas na dosažení 80% t [s]
86     TargTimeCold : REAL := 3.29; // Čas na dosažení 80% k [s]
87 END_VAR
88 IF NOT(INIT) THEN
89     TimeStep := TaskTime / 1000.0; // ms => s
90 END_IF;
91
92 (* Výpočet objemu, hmotnosti, energie *)
93 IF NOT(INIT) THEN
94     VAir := lenght * width * height;
95     MAir := RoAir * VAir;
96     QAir := MAir * CpAir;
97 END_IF;
98
99 (* Výpočet ploch *)
100 IF NOT(INIT) THEN
101     AreaWall1 := height * width;
102     AreaWall2 := height * width;
103     AreaWall3 := height * lenght;
104     AreaWall4 := height * lenght;
105     AreaFloor := lenght * width;
106     AreaCeiling := lenght * width;
107 END_IF;
108
109 (* Výpočet deltaT *)
110 DeltaTempWall_1 := wall_temp1 - RoomTemp;
111 DeltaTempWall_2 := wall_temp2 - RoomTemp;
112 DeltaTempWall_3 := wall_temp3 - RoomTemp;
113 DeltaTempWall_4 := wall_temp4 - RoomTemp;

```

```

114 DeltaTempFloor := floor_temp - RoomTemp;
115 DeltaTempCeiling := ceiling_temp - RoomTemp;
116
117 (* Ošetření teploty *)
118 (* Výpočet tepelných toků přes stěny *)
119 FiWall_1 := (LambdaBrick * AreaWall1 * DeltaTempWall_1) / wall_thic1;
120 FiWall_2 := (LambdaBrick * AreaWall2 * DeltaTempWall_2) / wall_thic2;
121 FiWall_3 := (LambdaBrick * AreaWall3 * DeltaTempWall_3) / wall_thic3;
122 FiWall_4 := (LambdaBrick * AreaWall4 * DeltaTempWall_4) / wall_thic4;
123 FiFloor := (LambdaBrick * AreaFloor * DeltaTempFloor) / floor_thic;
124 FiCeiling := (LambdaBrick * AreaCeiling * DeltaTempCeiling)/ceiling_thic;
125
126 (* Výpočet korekčních členů *)
127 IF NOT(INIT) THEN
128     KHeatRise := (TimeRise / TargTimeHeat) - 1.0;
129     KColdRise := (TimeRise / TargTimeCold) - 1.0;
130     KHeatFall1 := (LN(Heat_1_WATTS/Epsilon))/TimeFallHeat;
131     KHeatFall2 := (LN(Heat_2_WATTS/Epsilon))/TimeFallHeat;
132     KColdFall1 := (LN(Cold_1_WATTS/Epsilon))/TimeFallCold;
133     KColdFall2 := (LN(Cold_2_WATTS/Epsilon))/TimeFallCold;
134 END_IF;
135
136 (* Tepelný výkon topení a klimatizace *)
137 (* Výpočet alfy *)
138 IF NOT(INIT) THEN
139     AlphaHeat := LN(1 + KHeatRise * (TimeStep)) / LN(1 + KHeatRise *
140         ↪ TimeRise);
140     AlphaCold := LN(1 + KColdRise * (TimeStep)) / LN(1 + KColdRise *
141         ↪ TimeRise);
141 END_IF;
142
143 (* Topení 1 *)
144 IF Heat_1 THEN
145     FiHeatTmp1 := FiHeatTmp1 + AlphaHeat * (Heat_1_WATTS - FiHeatTmp1);
146 ELSE
147     FiHeatTmp1 := FiHeatTmp1 * EXP(-KHeatFall1 * (TimeStep));
148 END_IF;
149 (* Topení 2 *)
150 IF Heat_2 THEN
151     FiHeatTmp2 := FiHeatTmp2 + AlphaHeat * (Heat_2_WATTS - FiHeatTmp2);

```

```

152 ELSE
153     FiHeatTmp2 := FiHeatTmp2 * EXP(-KHeatFall2 * (TimeStep));
154 END_IF;
155 (* Klimatizace 1 *)
156 IF Cold_1 THEN
157     FiColdTmp1 := FiColdTmp1 + AlphaCold * (Cold_1_WATTS - FiColdTmp1);
158 ELSE
159     FiColdTmp1 := FiColdTmp1 * EXP(-KColdFall1 * (TimeStep));
160 END_IF;
161 (* Klimatizace 2 *)
162 IF Cold_2 THEN
163     FiColdTmp2 := FiColdTmp2 + AlphaCold * (Cold_2_WATTS - FiColdTmp2);
164 ELSE
165     FiColdTmp2 := FiColdTmp2 * EXP(-KColdFall2 * (TimeStep));
166 END_IF;
167
168 (* Suma výkonů *)
169 FiHeat := FiHeatTmp1 + FiHeatTmp2;
170 FiCold := FiColdTmp1 + FiColdTmp2;
171
172 (* Celkový tepelný tok *)
173 FiTotal := FiHeat - FiCold + FiWall_1 + FiWall_2 + FiWall_3 + FiWall_4 +
    ↪ FiFloor + FiCeiling;
174
175 (* Výpočet přírůstku teploty za task *)
176 DeltaTemp := (FiTotal / QAir) * TimeStep;
177 RoomTemp := RoomTemp + DeltaTemp;
178
179 (* Výstup *)
180 Temperature := RoomTemp;
181
182 (* NOT INIT *)
183 INIT := TRUE;
184 END_FUNCTION_BLOCK

```

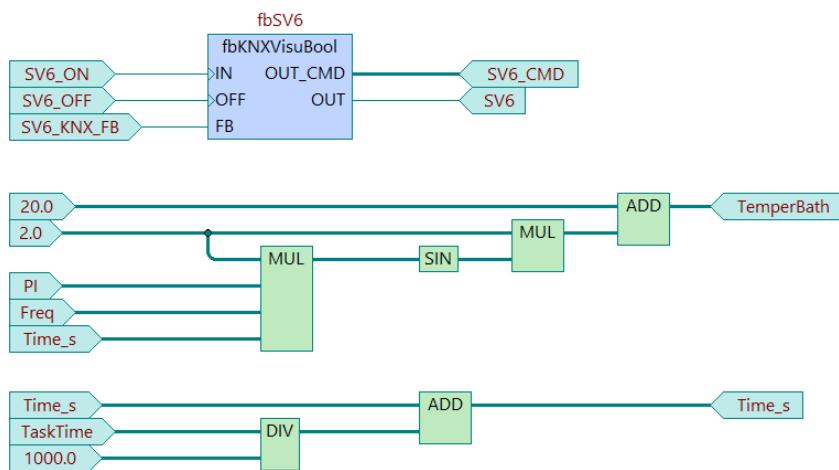
C.0.2 Popis funkčního bloku fbBath

Výpis C.2: fbBath

```

1  FUNCTION_BLOCK fbBath
2  VAR_INPUT
3      SV6_ON      : BOOL; //Vizu světlo 6 on
4      SV6_OFF     : BOOL; //Vizu světlo 6 off
5      SV6_KNX_FB  : BOOL; //KNX světlo 6 feedback
6  END_VAR
7  VAR_OUTPUT
8      SV6          : BOOL;    //Vizualizace světla 6
9      SV6_CMD      : DT_CMD_BOOL; //Příkaz světla 6
10     TemperBath   : REAL;    //Vizualizace + komunikace
11  END_VAR
12  VAR
13     Time_s       : REAL := 0.0;
14     fbSV6 : fbKNXVisuBool;
15  END_VAR
16  VAR CONSTANT
17     PI           : REAL := 3.14159265;
18     Freq         : REAL := 0.005; // perioda cca 3 minuty 20 sekund
19     TaskTime    : REAL := 400.0; // 250 ms
20  END_VAR

```



Obr. C.1: Logika funkčního bloku fbBath

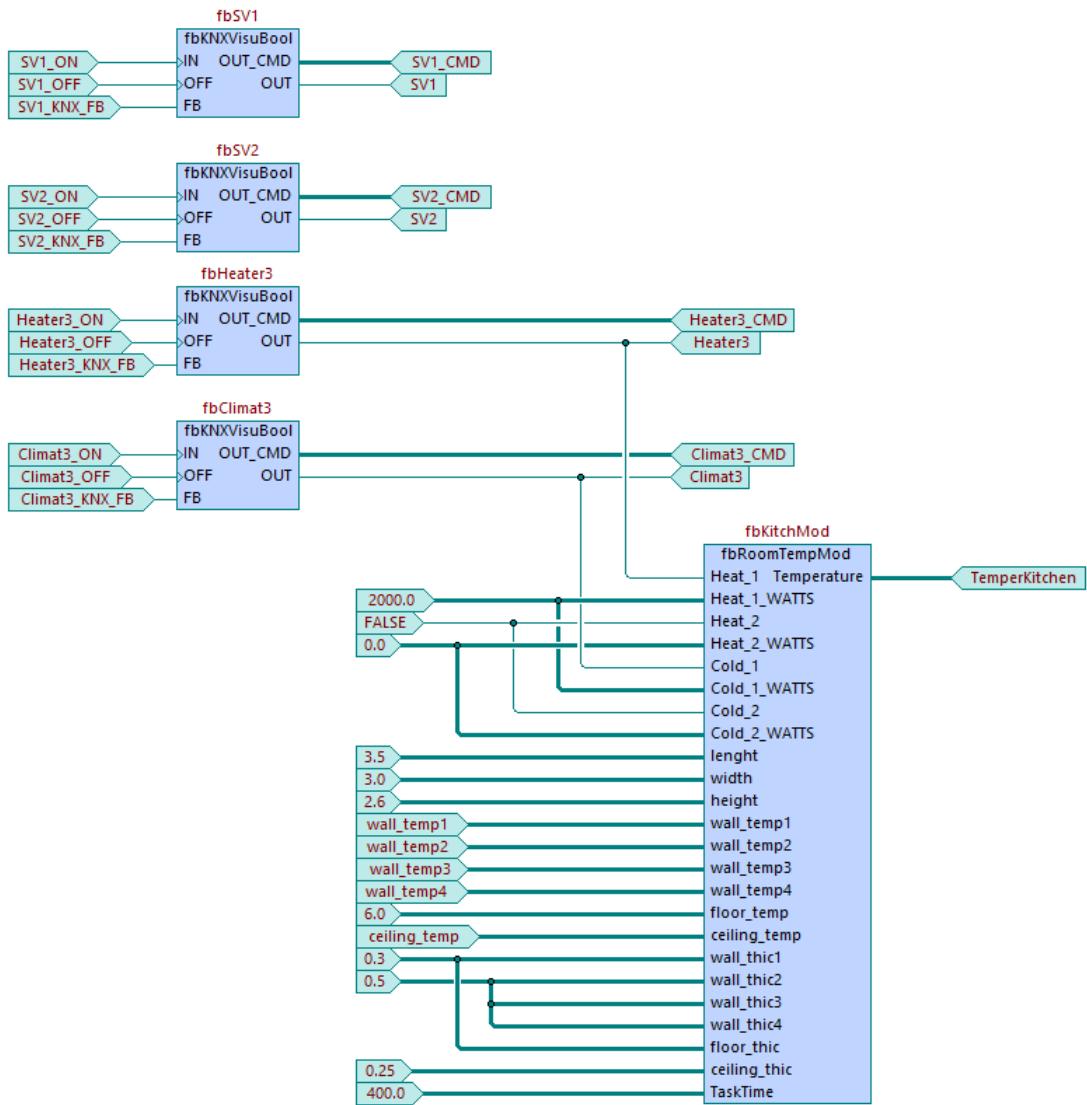
C.0.3 fbKitch

Výpis C.3: fbKitch

```

1  FUNCTION_BLOCK fbKitch
2
3  VAR_INPUT
4      SV1_ON          : BOOL; //Vizu světlo 1 on
5      SV1_OFF         : BOOL; //Vizu světlo 1 off
6      SV1_KNX_FB     : BOOL; //KNX světlo 1 feedback,
7      SV2_ON          : BOOL; //Vizu světlo 2 on
8      SV2_OFF         : BOOL; //Vizu světlo 2 off
9      SV2_KNX_FB     : BOOL; //KNX světlo 2 feedback
10     Heater3_ON     : BOOL; //Vizu toopení 3 on
11     Heater3_OFF    : BOOL; //Vizu toopení 3 off
12     Heater3_KNX_FB : BOOL; //KNX klimatizace 3 feedback
13     Climat3_ON      : BOOL; //Vizu klimatizace 3 on
14     Climat3_OFF     : BOOL; //Vizu klimatizace 3 off
15     Climat3_KNX_FB : BOOL; //KNX toopení 3 feedback
16     wall_temp1       : REAL; //Teplota koupelna[degC]
17     wall_temp2       : REAL; //Teplota ven[degC]
18     wall_temp3       : REAL; //Teplota ven[degC]
19     wall_temp4       : REAL; //Teplota ven[degC]
20     ceiling_temp     : REAL; //Teplota obyvak[degC]
21
22  END_VAR
23
24  VAR_OUTPUT
25      SV1           : BOOL; //Vizualizace světla 1
26      SV2           : BOOL; //Vizualizace světla 2
27      Heater3        : BOOL; //Vizualizace toopení 3
28      Climat3        : BOOL; //Vizualizace klimatizace 3
29      SV1_CMD        : DT_CMD_BOOL; //Příkaz světla 1
30      SV2_CMD        : DT_CMD_BOOL; //Příkaz světla 2
31      Heater3_CMD    : DT_CMD_BOOL; //Příkaz toopení 3
32      Climat3_CMD    : DT_CMD_BOOL; //Příkaz klimatizace 3
33      TemperKitchen  : REAL; //Vizualizace + komunikace
34
35  END_VAR
36
37  VAR
38      fbSV1 : fbKNXVisuBool;
39      fbSV2 : fbKNXVisuBool;
40      fbHeater3 : fbKNXVisuBool;
41      fbKitchMod : fbRoomTempMod;
42      fbClimat3 : fbKNXVisuBool;
43
44  END_VAR

```



Obr. C.2: Logika funkčního bloku **fbKitch**

C.0.4 fbLivRoom

Výpis C.4: fbLivRoom

```

1 FUNCTION_BLOCK fbLivRoom
2   VAR_INPUT
3     SV3_ON          : BOOL; //Vizu světlo 3 on
4     SV3_OFF         : BOOL; //Vizu světlo 3 off
5     SV3_KNX_FB      : BOOL; //KNX světlo 3 feedback,
6     SV4_ON          : BOOL; //Vizu světlo 4 on
7     SV4_OFF         : BOOL; //Vizu světlo 4 off
8     SV4_KNX_FB      : BOOL; //KNX světlo 4 feedback
9     SV5_ON          : BOOL; //Vizu světlo 5 on
10    SV5_OFF         : BOOL; //Vizu světlo 5 off
11    SV5_KNX_FB      : BOOL; //KNX světlo 5 feedback
12    Shut1_UP        : BOOL; //Vizu rolety 1 on
13    Shut1_DW        : BOOL; //Vizu rolety 1 down
14    Shut1_STEP_UP   : BOOL; //Vizu rolety 1 on krok
15    Shut1_STEP_DW   : BOOL; //Vizu rolety 1 off krok
16    Shut1_KNX_FB_UP : BOOL; //KNX rolety 1 feedback up
17    Shut1_KNX_FB_DW : BOOL; //KNX rolety 1 feedback down
18    Shut2_UP        : BOOL; //Vizu rolety 2 on
19    Shut2_DW        : BOOL; //Vizu rolety 2 down
20    Shut2_STEP_UP   : BOOL; //Vizu rolety 2 on krok
21    Shut2_STEP_DW   : BOOL; //Vizu rolety 2 off krok
22    Shut2_KNX_FB_UP : BOOL; //KNX rolety 2 feedback up
23    Shut2_KNX_FB_DW : BOOL; //KNX rolety 2 feedback down
24    Heater1_ON      : BOOL; //Vizu topení 1 on
25    Heater1_OFF     : BOOL; //Vizu topení 1 off
26    Heater1_KNX_FB : BOOL; //KNX klimatizace 1 feedback
27    Heater2_ON      : BOOL; //Vizu topení 2 on
28    Heater2_OFF     : BOOL; //Vizu topení 2 off
29    Heater2_KNX_FB : BOOL; //KNX klimatizace 2 feedback
30    Climat1_ON      : BOOL; //Vizu klimatizace 1 on
31    Climat1_OFF     : BOOL; //Vizu klimatizace 1 off
32    Climat1_KNX_FB : BOOL; //KNX topení 1 feedback
33    Climat2_ON      : BOOL; //Vizu klimatizace 2 on
34    Climat2_OFF     : BOOL; //Vizu klimatizace 2 off
35    Climat2_KNX_FB : BOOL; //KNX topení 2 feedback
36    wall_temp1       : REAL; // Teploplota koupelna[degC]
37    wall_temp2       : REAL; // Teploplota ven[degC]
38    wall_temp3       : REAL; // Teploplota ven[degC]
```

```

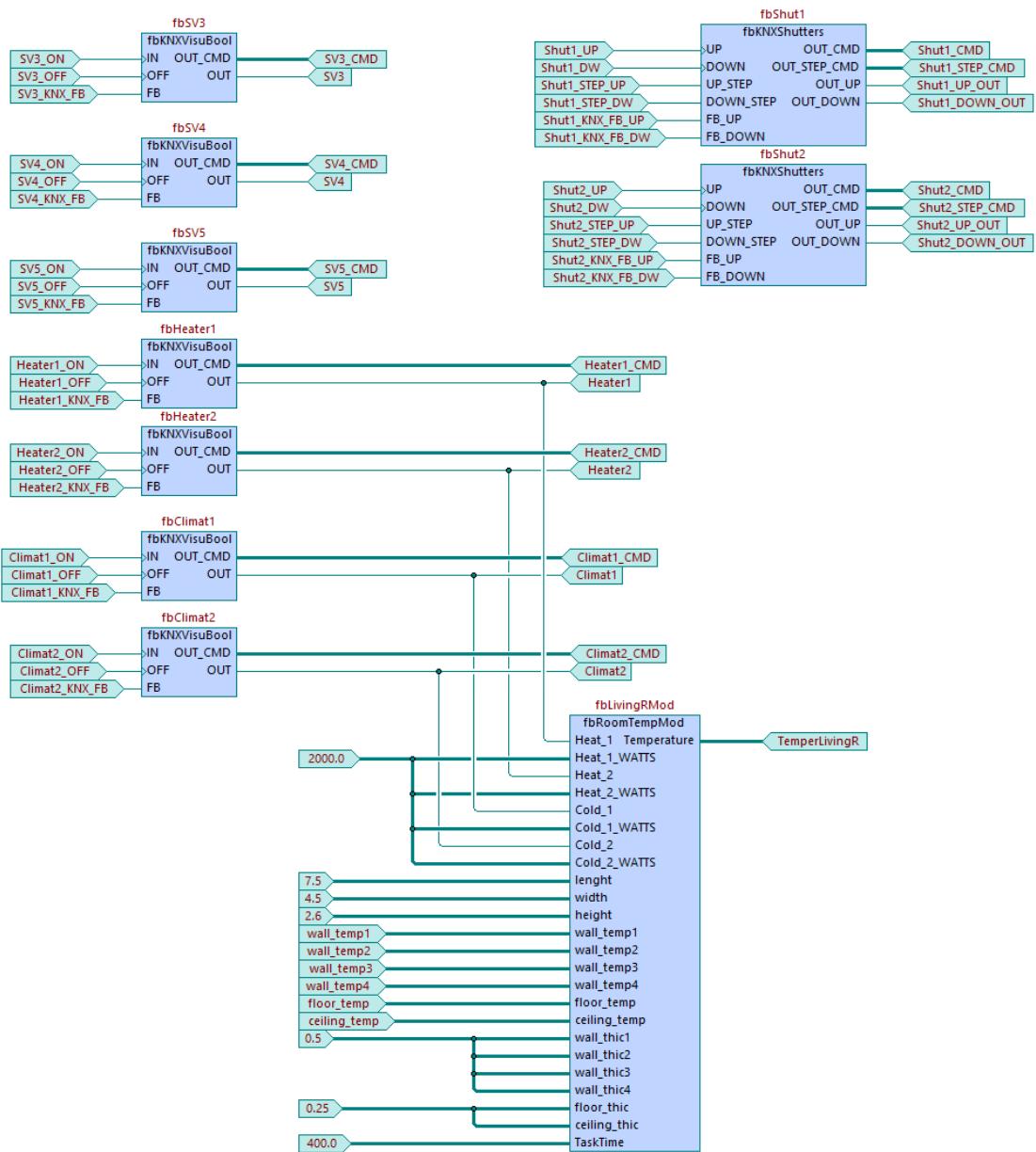
39    wall_temp4      : REAL; // Teplota ven[degC]
40    floor_temp      : REAL; // Teplota v místnosti pod[degC]
41    ceiling_temp    : REAL; // Teplota obyvák[degC]
42 END_VAR
43 VAR_OUTPUT
44    SV3            : BOOL;   //Vizualizace světla 3
45    SV4            : BOOL;   //Vizualizace světla 4
46    SV5            : BOOL;   //Vizualizace světla 5
47    SV3_CMD        : DT_CMD_BOOL; //Příkaz světla 3
48    SV4_CMD        : DT_CMD_BOOL; //Příkaz světla 4
49    SV5_CMD        : DT_CMD_BOOL; //Příkaz světla 5
50    Heater1        : BOOL;   //Vizualizace to opení 1
51    Heater2        : BOOL;   //Vizualizace to opení 2
52    Heater1_CMD    : DT_CMD_BOOL; //Příkaz to opení 1
53    Heater2_CMD    : DT_CMD_BOOL; //Příkaz to opení 2
54    Climat1        : BOOL;   //Vizualizace klimatizace 1
55    Climat2        : BOOL;   //Vizualizace klimatizace 2
56    Climat1_CMD    : DT_CMD_BOOL; //Příkaz klimatizace 1
57    Climat2_CMD    : DT_CMD_BOOL; //Příkaz klimatizace 2
58    Shut1_UP_OUT   : BOOL;   //Vizualizace žaluzie 1 UP
59    Shut1_DOWN_OUT : BOOL;   //Vizualizace žaluzie 1 DOWN
60    Shut1_CMD       : DT_CMD_BOOL; //Příkaz žaluzie 1
61    Shut1_STEP_CMD  : DT_CMD_BOOL; //Příkaz žaluzie 1 KROK
62    Shut2_UP_OUT   : BOOL;   //Vizualizace žaluzie 2 UP
63    Shut2_DOWN_OUT : BOOL;   //Vizualizace žaluzie 2 DOWN
64    Shut2_CMD       : DT_CMD_BOOL; //Příkaz žaluzie 2
65    Shut2_STEP_CMD  : DT_CMD_BOOL; //Příkaz žaluzie 2 KROK
66    TemperLivingR  : REAL;   //Vizualizace + komunikace
67 END_VAR
68 VAR
69    fbSV3 : fbKNXVisuBool;
70    fbSV4 : fbKNXVisuBool;
71    fbSV5 : fbKNXVisuBool;
72    fbHeater1 : fbKNXVisuBool;
73    fbClimat1 : fbKNXVisuBool;
74    fbHeater2 : fbKNXVisuBool;
75    fbClimat2 : fbKNXVisuBool;
76    fbLivingRMod : fbRoomTempMod;

```

```

77     fbShut1 : fbKNXShutters;
78     fbShut2 : fbKNXShutters;
79 END_VAR

```

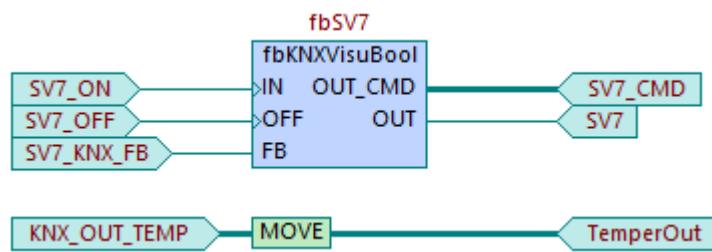


Obr. C.3: Logika funkčního bloku `fbLivRoom`

C.0.5 fbOutz

Výpis C.5: fbOutz

```
1  FUNCTION_BLOCK fbOutz
2
3      VAR_INPUT
4          SV7_ON      : BOOL R_EDGE; //Vizu světlo 7 on
5          SV7_OFF     : BOOL R_EDGE; //Vizu světlo 7 off
6          SV7_KNX_FB  : BOOL; //KNX světlo 7 feedback
7          KNX_OUT_TEMP : REAL; // KNX Teplota venku
8
9      END_VAR
10
11     VAR_OUTPUT
12         SV7          : BOOL;    //Vizualizace světla 7
13         SV7_CMD      : DT_CMD_BOOL; //Příkaz světla 7
14         TemperOut    : REAL;    //Vizualizace + komunikace
15
16     END_VAR
17
18     VAR
19         fbSV7 : fbKNXVisuBool;
20
21     END_VAR
```



Obr. C.4: Logika funkčního bloku fbOutz

D Program komunikace mezi PLC a KNX

Výpis D.1: Program komunikace mezi PLC a KNX

```
1 PROGRAM prgKNXComm
2 VAR
3     init : BOOL;
4     knx : fbKnxIpBaosBin;
5     knxObjectList : ARRAY[1..34] OF UDINT; // pole adres
6     SHUT1_FB_PULSE : TON;
7     SHUT2_FB_PULSE : TON;
8     datapoint1 : T_KNX_OBJECT_DPT1; // SV1_FB
9     datapoint2 : T_KNX_OBJECT_DPT1; // SV2_FB
10    datapoint3 : T_KNX_OBJECT_DPT1; // SV3_FB
11    datapoint4 : T_KNX_OBJECT_DPT1; // SV4_FB
12    datapoint5 : T_KNX_OBJECT_DPT1; // SV5_FB
13    datapoint6 : T_KNX_OBJECT_DPT1; // SV6_FB
14    datapoint7 : T_KNX_OBJECT_DPT1; // SV7_FB
15    datapoint8 : T_KNX_OBJECT_DPT1; // HEAT1_FB
16    datapoint9 : T_KNX_OBJECT_DPT1; // HEAT2_FB
17    datapoint10 : T_KNX_OBJECT_DPT1; // HEAT3_FB
18    datapoint11 : T_KNX_OBJECT_DPT1; // COLD1_FB
19    datapoint12 : T_KNX_OBJECT_DPT1; // COLD2_FB
20    datapoint13 : T_KNX_OBJECT_DPT1; // COLD3_FB
21    datapoint14 : T_KNX_OBJECT_DPT1; // SHUT1_FB
22    datapoint15 : T_KNX_OBJECT_DPT1; // SHUT1_CMD
23    datapoint16 : T_KNX_OBJECT_DPT1; // SHUT2_FB
24    datapoint17 : T_KNX_OBJECT_DPT1; // SHUT2_CMD
25    datapoint18 : T_KNX_OBJECT_DPT1; // SV1_CMD
26    datapoint19 : T_KNX_OBJECT_DPT1; // SV2_CMD
27    datapoint20 : T_KNX_OBJECT_DPT1; // SV3_CMD
28    datapoint21 : T_KNX_OBJECT_DPT1; // SV4_CMD
29    datapoint22 : T_KNX_OBJECT_DPT1; // SV5_CMD
30    datapoint23 : T_KNX_OBJECT_DPT1; // SV6_CMD
31    datapoint24 : T_KNX_OBJECT_DPT1; // SV7_CMD
32    datapoint25 : T_KNX_OBJECT_DPT1; // HEAT1_CMD
33    datapoint26 : T_KNX_OBJECT_DPT1; // HEAT2_CMD
34    datapoint27 : T_KNX_OBJECT_DPT1; // HEAT3_CMD
35    datapoint28 : T_KNX_OBJECT_DPT1; // COLD1_CMD
36    datapoint29 : T_KNX_OBJECT_DPT1; // COLD2_CMD
37    datapoint30 : T_KNX_OBJECT_DPT1; // COLD3_CMD
```

```

38 datapoint31      : T_KNX_OBJECT_DPT1; // krok Ž 1
39 datapoint32      : T_KNX_OBJECT_DPT1; // krok Ž 2
40 datapoint33      : T_KNX_OBJECT_DPT18; // scéna
41 datapoint34      : T_KNX_OBJECT_DPT9; // teplota
42 END_VAR
43 IF NOT init THEN // pole adres
44   knxObjectList[1] := PTR_TO_UDINT( ADR(datapoint1));
45   knxObjectList[2] := PTR_TO_UDINT( ADR(datapoint2));
46   knxObjectList[3] := PTR_TO_UDINT( ADR(datapoint3));
47   knxObjectList[4] := PTR_TO_UDINT( ADR(datapoint4));
48   knxObjectList[5] := PTR_TO_UDINT( ADR(datapoint5));
49   knxObjectList[6] := PTR_TO_UDINT( ADR(datapoint6));
50   knxObjectList[7] := PTR_TO_UDINT( ADR(datapoint7));
51   knxObjectList[8] := PTR_TO_UDINT( ADR(datapoint8));
52   knxObjectList[9] := PTR_TO_UDINT( ADR(datapoint9));
53   knxObjectList[10] := PTR_TO_UDINT( ADR(datapoint10));
54   knxObjectList[11] := PTR_TO_UDINT( ADR(datapoint11));
55   knxObjectList[12] := PTR_TO_UDINT( ADR(datapoint12));
56   knxObjectList[13] := PTR_TO_UDINT( ADR(datapoint13));
57   knxObjectList[14] := PTR_TO_UDINT( ADR(datapoint14));
58   knxObjectList[15] := PTR_TO_UDINT( ADR(datapoint15));
59   knxObjectList[16] := PTR_TO_UDINT( ADR(datapoint16));
60   knxObjectList[17] := PTR_TO_UDINT( ADR(datapoint17));
61   knxObjectList[18] := PTR_TO_UDINT( ADR(datapoint18));
62   knxObjectList[19] := PTR_TO_UDINT( ADR(datapoint19));
63   knxObjectList[20] := PTR_TO_UDINT( ADR(datapoint20));
64   knxObjectList[21] := PTR_TO_UDINT( ADR(datapoint21));
65   knxObjectList[22] := PTR_TO_UDINT( ADR(datapoint22));
66   knxObjectList[23] := PTR_TO_UDINT( ADR(datapoint23));
67   knxObjectList[24] := PTR_TO_UDINT( ADR(datapoint24));
68   knxObjectList[25] := PTR_TO_UDINT( ADR(datapoint25));
69   knxObjectList[26] := PTR_TO_UDINT( ADR(datapoint26));
70   knxObjectList[27] := PTR_TO_UDINT( ADR(datapoint27));
71   knxObjectList[28] := PTR_TO_UDINT( ADR(datapoint28));
72   knxObjectList[29] := PTR_TO_UDINT( ADR(datapoint29));
73   knxObjectList[30] := PTR_TO_UDINT( ADR(datapoint30));
74   knxObjectList[31] := PTR_TO_UDINT( ADR(datapoint31));
75   knxObjectList[32] := PTR_TO_UDINT( ADR(datapoint32));
76   knxObjectList[33] := PTR_TO_UDINT( ADR(datapoint33));
77   knxObjectList[34] := PTR_TO_UDINT( ADR(datapoint34));

```

```

78     init := TRUE;
79 END_IF
80 knx( firstKnxObject := 1,
81       lastKnxObject := 34,
82       ethCode := ETH2_uni2,
83       knxIP := STRING_TO_IPADR('192.168.xxx.xx'),
84       knxList := void( knxObjectList));
85
86 // Feedbacky
87 SV1_FB      := datapoint1 .value;
88 SV2_FB      := datapoint2 .value;
89 SV3_FB      := datapoint3 .value;
90 SV4_FB      := datapoint4 .value;
91 SV5_FB      := datapoint5 .value;
92 SV6_FB      := datapoint6 .value;
93 SV7_FB      := datapoint7 .value;
94 HEAT1_FB    := datapoint8 .value;
95 HEAT2_FB    := datapoint9 .value;
96 HEAT3_FB    := datapoint10.value;
97 COLD1_FB    := datapoint11.value;
98 COLD2_FB    := datapoint12.value;
99 COLD3_FB    := datapoint13.value;
100 SHUT1_FB_PULSE(IN := datapoint14.altValue, PT := T#1s);
101 SHUT2_FB_PULSE(IN := datapoint16.altValue, PT := T#1s);
102 SHUT1_FB_UP   := datapoint14.value AND SHUT1_FB_PULSE.Q;
103 SHUT1_FB_DOWN := NOT(datapoint14.value) AND SHUT1_FB_PULSE.Q;
104 SHUT2_FB_UP   := datapoint16.value AND SHUT2_FB_PULSE.Q;
105 SHUT2_FB_DOWN := NOT(datapoint16.value) AND SHUT2_FB_PULSE.Q;
106
107 // Příkazy
108 IF SV1_CMD.CMD      THEN datapoint18.value := SV1_CMD.CMD_VAL; END_IF;
109 IF SV2_CMD.CMD      THEN datapoint19.value := SV2_CMD.CMD_VAL; END_IF;
110 IF SV3_CMD.CMD      THEN datapoint20.value := SV3_CMD.CMD_VAL; END_IF;
111 IF SV4_CMD.CMD      THEN datapoint21.value := SV4_CMD.CMD_VAL; END_IF;
112 IF SV5_CMD.CMD      THEN datapoint22.value := SV5_CMD.CMD_VAL; END_IF;
113 IF SV6_CMD.CMD      THEN datapoint23.value := SV6_CMD.CMD_VAL; END_IF;
114 IF SV7_CMD.CMD      THEN datapoint24.value := SV7_CMD.CMD_VAL; END_IF;
115 IF HEAT1_CMD.CMD    THEN datapoint25.value := HEAT1_CMD.CMD_VAL; END_IF;
116 IF HEAT2_CMD.CMD    THEN datapoint26.value := HEAT2_CMD.CMD_VAL; END_IF;
117 IF HEAT3_CMD.CMD    THEN datapoint27.value := HEAT3_CMD.CMD_VAL; END_IF;

```

```

118 IF COLD1_CMD.CMD      THEN datapoint28.value := COLD1_CMD.CMD_VAL; END_IF;
119 IF COLD2_CMD.CMD      THEN datapoint29.value := COLD2_CMD.CMD_VAL; END_IF;
120 IF COLD3_CMD.CMD      THEN datapoint30.value := COLD3_CMD.CMD_VAL; END_IF;
121 IF SHUT1_CMD.CMD      THEN datapoint15.value := SHUT1_CMD.CMD_VAL; END_IF;
122 IF SHUT1_STEP_CMD.CMD THEN datapoint31.value := SHUT1_STEP_CMD.CMD_VAL;
   ↪ END_IF;
123 IF SHUT2_CMD.CMD      THEN datapoint17.value := SHUT1_CMD.CMD_VAL; END_IF;
124 IF SHUT2_STEP_CMD.CMD THEN datapoint32.value := SHUT1_STEP_CMD.CMD_VAL;
   ↪ END_IF;
125 // Shutdown scéna
126 IF SHUTDOWN_MQTT THEN
127   datapoint33.control := TRUE;
128   datapoint33.scene   := 5;
129 END_IF
130 // Posílání teploty
131 KNX_TEMPER := datapoint34.value;
132 END_PROGRAM

```

E Program komunikace mezi PLC a Home Assistantem

Výpis E.1: Program komunikace mezi PLC a Home Assistant – MQTT

```
1 PROGRAM prgMQTTComm
2
3     VAR
4
5         Pub      : fbMQTTPublisher;
6         Subs     : fbMQTTSubscriber;
7
8
9         brokerIPAddr : STRING := '192.168.xxx.xxx';
10        brokerPort   : UINT    := 1883;
11        localPort    : UINT    := 60000;
12        localPort2   : UINT    := 50000;
13        keepAlive    : BOOL    := TRUE;
14        keepAliveInterval : TIME  := T#60s;
15        pingInterval   : TIME  := T#10s;
16        connTimeOut    : TIME  := T#5s;
17
18
19        state : USINT := 0;
20        connMessage : STRING := 'Connected';
21        pubConnTopicMsg : STRING := 'plc/Connect/Publisher';
22        subConnTopicMsg : STRING := 'plc/Connect/Subscriber';
23        dataTopicMsg : STRING := 'plc/ALL_TEMPS';
24        pubComParam   : T_MQTT_COM_PUB_PARAM := (
25            pRetain := TRUE,
26            qos     := 1,
27            dup     := FALSE,
28            clean   := FALSE
29        );
30        willParamPub  : T_MQTT_COM_WILL_PARAM := (
31            wRetain := FALSE,
32            topic   := 'plc/Connect/Publisher',
33            mess    := 'Disconnected',
34            flag    := TRUE,
35            qos     := 1
36        );
37        subComParam   : T_MQTT_COM_SUB_PARAM := (
38            qos     := 1,
39            clean   := FALSE
40        );
41
```

```

37 willParamSub : T_MQTT_COM_WILL_PARAM := (
38     wRetain := FALSE,
39     topic := 'plc/Connect/Subscriber',
40     mess := 'Disconnected',
41     flag := TRUE,
42     qos := 1
43 );
44
45 jsonPayload : STRING[255];
46 sOut : STRING[16];
47 sKit : STRING[16];
48 sLiv : STRING[16];
49 sBath : STRING[16];
50 currentTopic : STRING[80];
51 currentData : STRING[255];
52 currentSendCom : BOOL;
53
54 pubTimer : TON;
55 stateTimer : TON;
56 stateCooldown : TON;
57 END_VAR
58 sOut := REAL_TO_STRINGF(OUTSIDE_TEMPER, '%.2f');
59 sKit := REAL_TO_STRINGF(KITCHEN_TEMPER, '%.2f');
60 sLiv := REAL_TO_STRINGF(LIVINGR_TEMPER, '%.2f');
61 sBath := REAL_TO_STRINGF(BATHROOM_TEMPER, '%.2f');
62 jsonPayload := CONCAT(
63     '{',
64     '"OUTSIDE_TEMPER": ', sOut, ',',
65     '"KITCHEN_TEMPER": ', sKit, ',',
66     '"LIVINGR_TEMPER": ', sLiv, ',',
67     '"BATHROOM_TEMPER": ', sBath,
68     '}'
69 );
70
71 CASE state OF
72     0: // Inicializace
73         currentTopic := '';
74         currentData := '';
75         currentSendCom := FALSE;

```

```

76   IF Pub.connected_to_broker THEN
77     stateTimer(IN:=TRUE, PT:=T#5s);
78   IF stateTimer.Q THEN
79     currentTopic := pubConnTopicMsg;
80     currentState := connMessage;
81     currentSendCom := TRUE;
82     stateTimer(IN:=FALSE);
83     state := 1;
84   END_IF;
85 ELSE
86   stateTimer(IN:=FALSE);
87 END_IF;
88
89 1: // Publisher Connected
90 IF NOT Pub.busy AND NOT currentSendCom THEN
91   stateCooldown(IN:=TRUE, PT:=T#3s);
92   IF stateCooldown.Q THEN
93     currentTopic := subConnTopicMsg;
94     currentState := connMessage;
95     currentSendCom := TRUE;
96     stateCooldown(IN:=FALSE);
97     state := 2;
98   END_IF;
99 END_IF;
100
101 2: // Subs Connected
102 IF NOT Pub.busy AND NOT currentSendCom THEN
103   stateCooldown(IN:=TRUE, PT:=T#3s);
104   IF stateCooldown.Q THEN
105     stateCooldown(IN:=FALSE);
106     state := 3;
107     pubTimer(IN:=FALSE);
108   END_IF;
109 END_IF;
110
111 3: // Normální provoz
112 currentTopic := dataTopicMsg;
113 currentState := jsonPayload;
114 pubTimer(IN:=TRUE, PT:=T#5s);

```

```

115 IF pubTimer.Q AND NOT Pub.busy AND NOT currentSendCom THEN
116   currentSendCom := TRUE;
117   pubTimer(IN:=FALSE);
118 END_IF;
119
120 IF NOT Pub.connected_to_broker AND NOT Subs.connected_to_broker THEN
121   state := 0;
122 END_IF;
123 END_CASE;
124
125 Pub(
126   chanCode      := ETH2_uni1,
127   brokerIP      := STRING_TO_IPADR(brokerIPAddr),
128   brokerPort    := brokerPort,
129   localPort     := localPort,
130   connect       := TRUE,
131   keepAlive     := keepAlive,
132   keepAliveInterval := keepAliveInterval,
133   pingInterval  := pingInterval,
134   connTimeOut   := connTimeOut,
135   clientId_auto := FALSE,
136   clientId      := 'PLC_TECO_CP2007_PUB',
137   comParam      := pubComParam,
138   willParam     := willParamPub,
139   topicTxt      := currentTopic,
140   dataTxt       := currentData,
141   sendCom       := currentSendCom
142 );
143
144 IF currentSendCom AND Pub.busy THEN
145   currentSendCom := FALSE;
146 END_IF;
147
148 Subs(
149   chanCode      := ETH2_uni0,
150   brokerIP      := STRING_TO_IPADR(brokerIPAddr),
151   brokerPort    := brokerPort,
152   localPort     := localPort2,
153   connect       := TRUE,
154   keepAlive     := keepAlive,

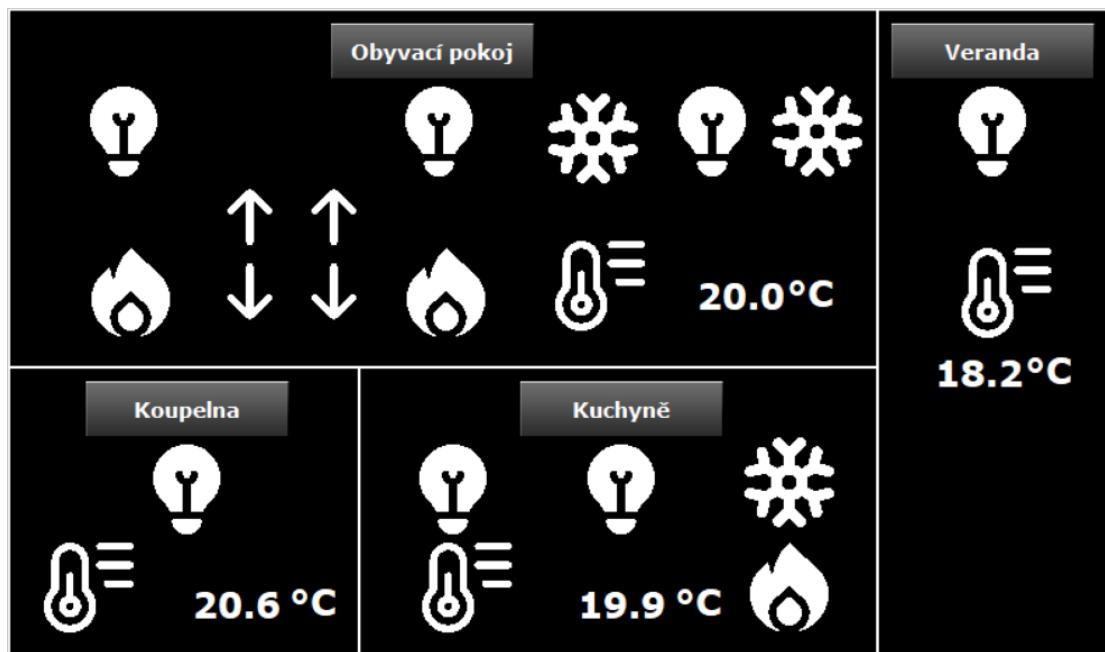
```

```

155     keepAliveInterval := keepAliveInterval,
156     pingInterval      := pingInterval,
157     connTimeOut       := connTimeOut,
158     clientId_auto    := FALSE,
159     clientId          := 'PLC_TECO_CP2007_SUB',
160     loginName         := '',
161     loginPass         := '',
162     comParam          := subComParam,
163     willParam         := willParamSub,
164     subRq              := Subs.connected_to_broker,
165     unSubRq            := FALSE,
166     topicTxt          := 'ha/SHUTDOWN'
167 );
168
169 IF Subs.dataRec THEN
170   SHUTDOWN_MQTT := (Subs.dataTxt = 'true');
171 ELSE
172   SHUTDOWN_MQTT := FALSE;
173 END_IF;
174 END_PROGRAM

```


F Vizualizace pomocí WebMakeru



Obr. F.1: WebMaker - přehled



Obr. F.2: WebMaker - Veranda



Obr. F.3: WebMaker -Kuchyně



Obr. F.4: WebMaker - Koupelna



Obr. F.5: WebMaker - Obývací pokoj

G Docker YAML soubory

G.0.1 Portainer YAML

Výpis G.1: Portainer YAML

```
1 services:
2   portainer:
3     image: portainer/portainer-ce:latest
4     container_name: portainer
5     ports:
6       - 9000:9000
7     volumes:
8       - portainer_data:/data
9       - /var/run/docker.sock:/var/run/docker.sock
10    restart: unless-stopped
```

G.0.2 Stack YAML

Výpis G.2: Stack YAML

```
1 services:
2   home-assistant:
3     container_name: home-assistant
4     image: homeassistant/home-assistant:2025.4
5     volumes:
6       - /mnt/sda1/usb/config/home-assistant:/config
7       - /etc/localtime:/etc/localtime:ro
8       - /etc/timezone:/etc/timezone:ro
9       - /run/dbus:/run/dbus:ro
10    restart: unless-stopped
11    privileged: true
12    network_mode: host
13  influxdb:
14    container_name: influxdb
15    image: influxdb:2.7.11
16    ports:
17      - 8086:8086
18    restart: unless-stopped
19    volumes:
20      - /mnt/sda1/usb/data/influxdb:/var/lib/influxdb2
21      - /mnt/sda1/usb/config/influxdb:/etc/influxdb2
22      - /etc/localtime:/etc/localtime:ro
23      - /etc/timezone:/etc/timezone:ro
24    environment:
25      - INFLUXDB_DB=db0
26      - DOCKER_INFLUXDB_INIT_MODE=setup
27      - DOCKER_INFLUXDB_INIT_USERNAME=${INFLUXDB_INIT_USERNAME}
28      - DOCKER_INFLUXDB_INIT_PASSWORD=${INFLUXDB_INIT_PASSWORD}
29      - DOCKER_INFLUXDB_INIT_ORG=fekt
30      - DOCKER_INFLUXDB_INIT_BUCKET=kybliceck
31  grafana:
32    image: grafana/grafana-oss:11.6.0
33    container_name: grafana
34    user: "${UID_GRAFANA}:${GID_GRAFANA}"
35    ports:
36      - 3000:3000
37    restart: unless-stopped
```

```

38   volumes:
39     - /mnt/sda1/usb/data/grafana:/var/lib/grafana
40     - /etc/grafana/provisioning:/etc/grafana/provisioning
41     - /etc/localtime:/etc/localtime:ro
42     - /etc/timezone:/etc/timezone:ro
43   depends_on:
44     - influxdb
45   environment:
46     - GF_DATABASE_TIMEOUT=1000
47     - GF_DATABASE_MAX_OPEN_CONN=2
48     - GF_DATABASE_MAX_IDLE_CONN=2
49   mosquitto:
50     container_name: mosquitto
51     image: eclipse-mosquitto:2.0.21
52     restart: unless-stopped
53     user: "${UID_MOSQUITTO}:${GID_MOSQUITTO}"
54     ports:
55       - "1883:1883"
56       - "9001:9001"
57     volumes:
58       - /mnt/sda1/usb/config/mosquitto:/mosquitto/config
59       - /mnt/sda1/usb/data/mosquitto:/mosquitto/data
60       - /mnt/sda1/usb/logs/mosquitto:/mosquitto/log

```


H Home Assistant - konfigurační soubor

Výpis H.1: Home Assistant configuration.yaml

```
1 # Loads default set of integrations. Do not remove.
2 default_config:
3
4 # Load frontend themes from the themes folder
5 frontend:
6   themes: !include_dir_merge_named themes
7
8 automation: !include automations.yaml
9 script: !include scripts.yaml
10 scene: !include scenes.yaml
11
12 influxdb:
13   api_version: 2
14   ssl: false
15   host: localhost
16   port: 8086
17   token: xxx
18   organization: xxx
19   bucket: kyblicek
20   tags:
21     source: HA
22   tags_attributes:
23     - friendly_name
24   default_measurement: units
25
26 mqtt:
27   sensor:
28     - name: "Venek teplota"
29       state_topic: "plc/ALL_TEMPS"
30       value_template: "{{ value_json.OUTSIDE_TEMPER | float }}"
31       device_class: "temperature"
32       unit_of_measurement: "..C"
33       unique_id: "tmp_out"
34     - name: "Kuchyn teplota"
35       state_topic: "plc/ALL_TEMPS"
36       value_template: "{{ value_json.KITCHEN_TEMPER | float }}"
37       device_class: "temperature"
```

```

38    unit_of_measurement: "..C"
39    unique_id: "tmp_kitch"
40  - name: "Obyvak teplota"
41    state_topic: "plc/ALL_TEMPS"
42    value_template: "{{ value_json.LIVINGR_TEMPER | float }}"
43    device_class: "temperature"
44    unit_of_measurement: "..C"
45    unique_id: "tmp_liv"
46  - name: "Koupelna teplota"
47    state_topic: "plc/ALL_TEMPS"
48    value_template: "{{ value_json.BATHROOM_TEMPER | float }}"
49    device_class: "temperature"
50    unit_of_measurement: "..C"
51    unique_id: "tmp_bath"
52 binary_sensor:
53  - name: "PLC MQTT Publisher"
54    state_topic: "plc/Connect/Publisher"
55    payload_on: "Connected"
56    payload_off: "Disconnected"
57    device_class: connectivity
58    unique_id: plc_mqtt_publisher
59  - name: "PLC MQTT Subscriber"
60    state_topic: "plc/Connect/Subscriber"
61    payload_on: "Connected"
62    payload_off: "Disconnected"
63    device_class: connectivity
64    unique_id: plc_mqtt_subscriber
65 button:
66  - name: "Central stop"
67    unique_id: "MQTT_Shutdown"
68    command_topic: "ha/SHUTDOWN"
69    payload_press: "true"

```

I Nastavení přehledů v Grafaně

I.0.1 Grafana - Spotřeba

Výpis I.1: Grafana - Spotřeba klimatizace

```
1 from(bucket: "kyblicek")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Klimatizace u televize" or r
6     ↵ ["friendly_name"] == "Leva zaluzie" or r["friendly_name"] == "Prava
7     ↵ zaluzie")
8   |> map(fn: (r) => ({ r with _value: float(v: r._value) * 1000.0 }))
9   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
10  |> fill(column: "_value", usePrevious: true)
11  |> window(every: 1h, createEmpty: false)
12  |> integral(unit: 1h)
13  |> last()
14  |> group(columns: [])
15  |> sum(column: "_value")
16  |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
17  |> yield(name: "Spotreba klimatizace za posledni hodinu (kWh)")
```

Výpis I.2: Grafana - Spotřeba topení

```
1 from(bucket: "kyblicek")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Topeni kuchyne" or r[
6     ↵ friendly_name] == "Topeni pracovna" or r["friendly_name"] == "
7     ↵ Topeni u televize")
8   |> map(fn: (r) => ({ r with _value: float(v: r._value) * 4000.0 }))
9   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
10  |> fill(column: "_value", usePrevious: true)
11  |> window(every: 1h, createEmpty: false)
12  |> integral(unit: 1h)
13  |> last()
14  |> group(columns: [])
15  |> sum(column: "_value")
16  |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
17  |> yield(name: "Spotreba topeni za posledni hodinu (kWh)")
```

Výpis I.3: Grafana - Spotřeba světla

```

1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "light")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> map(fn: (r) => ({ r with _value: float(v: r._value) * 15.0 }))
6   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
7   |> fill(column: "_value", usePrevious: true)
8   |> window(every: 1h, createEmpty: false)
9   |> integral(unit: 1h)
10  |> last()
11  |> group(columns: [])
12  |> sum(column: "_value")
13  |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
14  |> yield(name: "Spotreba svetel za posledni hodinu (kWh)")

```

Výpis I.4: Grafana - Rozložení spotřeby

```

1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Klimatizace kuchyne" or r["
    ↪ friendly_name] == "Klimatizace pracovna" or r["friendly_name"] ==
    ↪ "Klimatizace u televize")
6   |> map(fn: (r) => ({ r with _value: float(v: r._value) * 1000.0 }))
7   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
8   |> fill(column: "_value", usePrevious: true)
9   |> window(every: 1h, createEmpty: false)
10  |> integral(unit: 1h)
11  |> last()
12  |> group(columns: [])
13  |> sum(column: "_value")
14  |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
15  |> yield(name: "Spotreba klimatizace za posledni hodinu (kWh)")
16 from(bucket: "kybliceck")
17   |> range(start: -1h)
18   |> filter(fn: (r) => r["domain"] == "switch")
19   |> filter(fn: (r) => r["_field"] == "value")
20   |> filter(fn: (r) => r["friendly_name"] == "Leva zaluzie" or r["
    ↪ friendly_name] == "Prava zaluzie")
21   |> map(fn: (r) => ({ r with _value: float(v: r._value) * 150.0 }))

```

```

22 |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
23 |> fill(column: "_value", usePrevious: true)
24 |> window(every: 1h, createEmpty: false)
25 |> integral(unit: 1h)
26 |> last()
27 |> group(columns: [])
28 |> sum(column: "_value")
29 |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
30 |> yield(name: "Spotreba zaluzii")
31 from(bucket: "kyblicek")
32 |> range(start: -1h)
33 |> filter(fn: (r) => r["domain"] == "switch")
34 |> filter(fn: (r) => r["_field"] == "value")
35 |> filter(fn: (r) => r["friendly_name"] == "Topeni kuchyne" or r[""
36   ↵ friendly_name"] == "Topeni pracovna" or r["friendly_name"] == ""
37   ↵ Topeni u televize")
38 |> map(fn: (r) => ({ r with _value: float(v: r._value) * 4000.0 }))
39 |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
40 |> fill(column: "_value", usePrevious: true)
41 |> window(every: 1h, createEmpty: false)
42 |> integral(unit: 1h)
43 |> last()
44 |> group(columns: [])
45 |> sum(column: "_value")
46 |> map(fn: (r) => ({ r with _value: r._value / 1000.0 }))
47 |> yield(name: "Spotreba openi za posledni hodinu (kWh)")
48 from(bucket: "kyblicek")
49 |> range(start: -1h)
50 |> filter(fn: (r) => r["domain"] == "light")
51 |> filter(fn: (r) => r["_field"] == "value")
52 |> map(fn: (r) => ({ r with _value: float(v: r._value) * 15.0 }))
53 |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
54 |> fill(column: "_value", usePrevious: true)
55 |> window(every: 1h, createEmpty: false)
56 |> integral(unit: 1h)
57 |> last()
58 |> group(columns: [])
59 |> sum(column: "_value")

```

I.0.2 Grafana - Spínání

Výpis I.5: Grafana - Spínání klimatizace

```
1 from(bucket: "kybliceck")
2   |> range(start: -24h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Klimatizace kuchyne" or r[
6     ↪ friendly_name] == "Klimatizace pracovna" or r["friendly_name"] ==
7     ↪ "Klimatizace u televize")
8   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
9   |> fill(column: "_value", usePrevious: true)
10  |> yield(name: "Stav klimatizaci kazdou minutu")
```

Výpis I.6: Grafana - Spínání topení

```
1 from(bucket: "kybliceck")
2   |> range(start: -24h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Topeni kuchyne" or r[
6     ↪ friendly_name] == "Topeni pracovna" or r["friendly_name"] == "
7     ↪ Topeni u televize")
8   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
9   |> fill(column: "_value", usePrevious: true)
10  |> yield(name: "Stav toopeni kazdou minutu")
```

Výpis I.7: Grafana - Spínání světel

```
1 from(bucket: "kybliceck")
2   |> range(start: -24h)
3   |> filter(fn: (r) => r["domain"] == "light")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
6   |> fill(column: "_value", usePrevious: true)
7   |> yield(name: "Stav svetel kazdou minutu")
```

Výpis I.8: Grafana - Spínání žaluzí

```
1 from(bucket: "kyblicek")
2   |> range(start: -24h)
3   |> filter(fn: (r) => r["domain"] == "switch")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["friendly_name"] == "Leva zaluzie" or r[
6     ↪ friendly_name] == "Prava zaluzie")
7   |> aggregateWindow(every: 1m, fn: last, createEmpty: true)
8   |> fill(column: "_value", usePrevious: true)
9   |> yield(name: "Stav zaluzii kazdou minutu")
```

I.0.3 Grafana - Celkové měsíční náklady

Výpis I.9: Grafana - Celkové měsíční náklady

```
1 price = 3.2
2 klimatizace = from(bucket: "kybliceck")
3 |> range(start: -30d)
4 |> filter(fn: (r) => r["domain"] == "switch")
5 |> filter(fn: (r) => r["_field"] == "value")
6 |> filter(fn: (r) => r["friendly_name"] == "Klimatizace kuchyne" or r[
7   ↵ friendly_name] == "Klimatizace pracovna" or r["friendly_name"] ==
8   ↵ "Klimatizace u televize")
9 |> map(fn: (r) => ({ r with _value: if r._value > 0.0 then 1.0 else 0.0
10   ↵ }))
11 |> aggregateWindow(every: 1h, fn: mean, createEmpty: false)
12 |> map(fn: (r) => ({ r with _value: r._value * 1.0 }))
13 |> sum()
14 |> map(fn: (r) => ({ r with _value: r._value * price, kategorie: "
15   ↵ Klimatizace" }))
16 toopeni = from(bucket: "kybliceck")
17 |> range(start: -30d)
18 |> filter(fn: (r) => r["domain"] == "switch")
19 |> filter(fn: (r) => r["_field"] == "value")
20 |> filter(fn: (r) => r["friendly_name"] == "Topeni kuchyne" or r[
21   ↵ friendly_name] == "Topeni pracovna" or r["friendly_name"] == "
22   ↵ Topeni u televize")
23 |> map(fn: (r) => ({ r with _value: if r._value > 0.0 then 1.0 else 0.0
24   ↵ }))
25 |> aggregateWindow(every: 1h, fn: mean, createEmpty: false)
26 |> map(fn: (r) => ({ r with _value: r._value * 4.0 }))
27 |> sum()
28 |> map(fn: (r) => ({ r with _value: r._value * price, kategorie: "Topeni
29   ↵ " }))
```

```

32  |> map(fn: (r) => ({ r with _value: r._value * 0.15 }))
33  |> sum()
34  |> map(fn: (r) => ({ r with _value: r._value * price, kategorie: "
35   ↪ Zaluzie" }))
36  svetla = from(bucket: "kyblicek")
37  |> range(start: -30d)
38  |> filter(fn: (r) => r["domain"] == "light")
39  |> filter(fn: (r) => r["_field"] == "value")
40  |> map(fn: (r) => ({ r with _value: if r._value > 0.0 then 1.0 else 0.0
41   ↪ })))
42  |> aggregateWindow(every: 1h, fn: mean, createEmpty: false)
43  |> map(fn: (r) => ({ r with _value: r._value * 0.015 }))
44  |> sum()
45  |> map(fn: (r) => ({ r with _value: r._value * price, kategorie: "Svetla
46   ↪ " }))
47  union(tables: [klimatizace, topeni, zaluzie, svetla]) // celkova sumu
48  |> group()
49  |> sum()
50  |> map(fn: (r) => ({ r with typ: "Celkove naklady" }))
51  |> yield(name: "Celkove mesicni naklady (Kc)")

```

I.0.4 Grafana - Průběh teploty

Výpis I.10: Grafana - Průběh teploty - Query A

```
1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "sensor")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["_measurement"] == "C")
6   |> filter(fn: (r) => r["entity_id"] == "venek_teplota")
7   |> aggregateWindow(every: 1m, fn: mean, createEmpty: false)
8   |> yield(name: "Teplota venku")
```

Výpis I.11: Grafana - Průběh teploty - Query B

```
1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "sensor")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["_measurement"] == "C")
6   |> filter(fn: (r) => r["entity_id"] == "kuchyn_teplota")
7   |> aggregateWindow(every: 1m, fn: mean, createEmpty: false)
8   |> yield(name: "Teplota v kuchyni")
```

Výpis I.12: Grafana - Průběh teploty - Query C

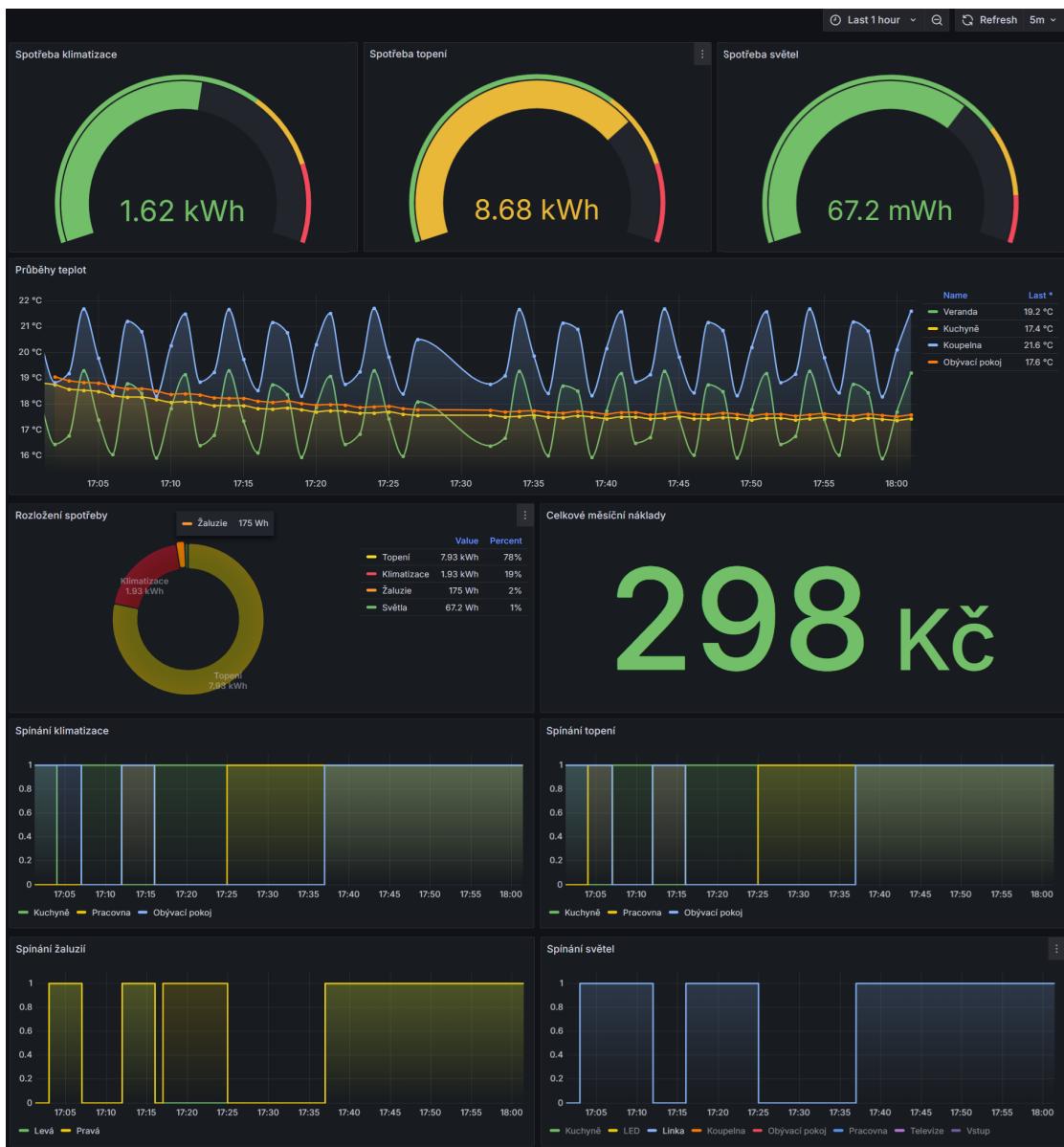
```
1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "sensor")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["_measurement"] == "C")
6   |> filter(fn: (r) => r["entity_id"] == "koupelna_teplota")
7   |> aggregateWindow(every: 1m, fn: mean, createEmpty: false)
8   |> yield(name: "Teplota v koupelne")
```

Výpis I.13: Grafana - Průběh teploty - Query D

```

1 from(bucket: "kybliceck")
2   |> range(start: -1h)
3   |> filter(fn: (r) => r["domain"] == "sensor")
4   |> filter(fn: (r) => r["_field"] == "value")
5   |> filter(fn: (r) => r["_measurement"] == "C")
6   |> filter(fn: (r) => r["entity_id"] == "obyvak_teplota")
7   |> aggregateWindow(every: 1m, fn: mean, createEmpty: false)
8   |> yield(name: "Teplota v obyvacim pokoji")

```



Obr. I.1: Vzhled přehledu v Grafaně