

GYMNÁZIUM A ZUŠ ŠLAPANICE



Intelligentní dům

Kobeřice u Brna

2020/2021

Hana Liškařová

ROČNÍKOVÁ PRÁCE

Obor: Informatika

Intelligentní dům

Smart house

Gymnázium a ZUŠ Šlapanice

VEDOUCÍ PRÁCE:

Mgr. Roman Ondrůšek

AUTOR:

Hana Liškařová

Kobeřice u Brna

2020/2021

Prohlášení

Prohlašuji, že jsem tuto ročníkovou práci vypracovala samostatně s použitím uvedené literatury a pramenů.

V Kobeřicích u Brna dne 31.1. 2021

Podpis Liškařová

Poděkování

Mé poděkování patří Mgr. Romanovi Ondrůškovi za odborné vedení, cenné rady a trpělivost, kterou mi během tvorby mé práce věnoval. Také chci poděkovat panu Liškaři, za zapůjčení řídicí jednotky a poskytnutí cenných informací. Dále bych chtěla poděkovat škole za možnost tvořit tuto práci.

Anotace

Práce se zabývá tvořením programu kontrolujícího inteligentní dům. Program je určen pro řídicí jednotku Amit a je napsán v jazyce aplikace DetStudio. Program je pro byt 2+kk a slouží k usnadnění a větší pohodlí životu v bytě. V textu práce je podrobně popsána tvorba každé části programu a co přesně má program ovládat.

Abstract¹

This seminar work deals with creating a program to control intelligent house. The program is designed for a control unit from Amit and is written in the programming language of DetStudio. The program is meant to be used for a 2+kk apartment and its purpose is to facilitate and increase the comfort of living in said apartment. The text of the thesis describes in detail the process of creating each part of the program and what exactly should each part control.

Klíčová slova

Ročníková práce, Gymnázium a ZUŠ Šlapanice, Inteligentní dům, Amit, Osvětlení, Podlahové topení, DetStudio, Pohybové čidlo, Teplotní čidlo, Program, Byt, Proměnná, Alias, Modul, 2+kk, Boiler

OBSAH

1 Teoretický úvod	7
1.1 Inteligentní dům	7
1.1.1 Pojem Inteligentní dům	7
1.1.2 Pojem Internet věcí	7
1.2 Možnosti automatizace	8
1.3 Kompatibilita	8
1.4 Možnost instalace	8
1.5 Spolehlivost a rizika	8
2 Praktická část	10
2.1 Popis výchozího stavu, definice cílového	10
2.2 Použité technologie	11
2.3 Tvorba programu	12
2.3.1 Výběr platformy, způsobu programování	12
2.3.2 Proměnné a Aliasy	12
2.3.3 Rozdělení do procesů	15
2.3.4 Tvorba jednotlivých procesů	15
2.3.4.1 rychle_p	15
2.3.4.2 Topení	24
2.3.4.3 vstupy_vy	26
2.4 Ladění programu	29
3 Závěr	29
4 Fotografie	31

1. Teoretický úvod

1.1 Inteligentní dům

Téma inteligentních domů jsem si zvolila, protože jsem přesvědčena, že je to téma co se bude řešit i v budoucnu. S narůstající popularitou inteligentních domů si můžu být jistá, když řeknu, že bych se chtěla tímto tématem zajímat i do budoucna. Inteligentní domy mě zajímají kvůli jejich možnostem. Každý dům může být jiný. Jeden se může například hlavně zaměřovat na úsporu energií a druhý na co největší automatizaci. S tím se pojí možnosti automatizace, kompatibilita i možnosti instalace. Stejně jako většinu lidí, mě trochu děsí rizika takovýchto domů. Ale především mě zajímá jak je celý inteligentní dům propojen. Jak funguje a jak se tvoří systém, co takovýto dům řídí. Jak se potom takovýto systém udržuje a jak je ho možné zabezpečit.

1.1.1 Pojem Inteligentní dům

Co že je to vlastně inteligentní dům? **Inteligentní dům** je takový dům, který zajišťuje optimální vnitřní prostředí pro komfort osob prostřednictvím stavební konstrukce, techniky prostředí, řídicích systémů, služeb a managementu. Je efektivní ekonomicky, energeticky i z hlediska působení na vnější prostředí a umožňuje víceúčelové použití a rekonfigurace. Reaguje na potřeby obyvatel s cílem zvýšit jejich pohodlí, zaručit co nejvyšší bezpečí a snížit náklady na provoz. ^[1]

S použitím mnoha různých zařízení přichází požadavek na jejich vzájemnou koordinaci a nový druh ovládání. Každé zařízení v sobě má řadu podsystémů, kterými výrobce optimalizoval výslednou funkci. Také jejich sestava, tvořící inteligentní dům, by měla být funkčně optimalizována. Inteligentní dům je jedno velké zařízení, jehož cílové funkce určuje jeho uživatel. Každý inteligentní dům je individuální, díky kombinaci typů zařízení od různých výrobců a specifickou logikou ovládacích prvků a také interiérem kterému se musel přizpůsobit.

1.1.2 Pojem Internet věcí

Internet věcí je označení pro síť fyzických zařízení, vozidel, domácích spotřebičů a dalších zařízení, která jsou vybavena elektronikou, softwarem, senzory, pohyblivými částmi a sítíovou konektivitou, která umožňuje těmto zařízením se propojit a vyměňovat si data. Každé z těchto

zařízení je jasně identifikovatelné díky implementovanému výpočetnímu systému, ale přesto je schopno pracovat samostatně v existující infrastruktuře internetu. ^[2]

1.2 Možnosti automatizace

Možností automatizace jsou spousty. Nejprve si budoucí uživatel musí určit, co přesně od inteligentního domu očekává, nejlépe se specializovaným projektantem. Systém musí být navržen tak, aby umožnil pohodlný způsob, jak ovládat všechny zařízení v domě dle potřeb uživatele. Uživatel musí pečlivě zvážit výběr značky řídicího systému (v našem případě Amit), i společnosti která bude tento systém realizovat tak, aby byla schopna obstát všem požadavkům. Většina značek se soustředí jen na malý okruh domácnosti (topení, ovládání světel...), čímž může omezit možnosti jejich použití v rámci plnější domácí automatizace.

1.3 Kompatibilita

Při nákupu zařízení musí být uživatel již rozhodnut, co přesně od svého inteligentního domu chce. Zařízení a systému je hodně, jsou specializované, ale žádný z nich není úplně univerzální. Uživatel by měl vědět, která zařízení komunikují se systémy používanými pro realizaci inteligentních domů. Někteří výrobci vybavují své zařízení komunikačními kanály, ale nezveřejňují jejich protokol. Pro jejich zapojení do systému pak dodávají komunikační moduly, jejichž cena je někdy překvapivě vysoká. Týká se to často známých značek. U některých systémů se ani nepředpokládá, že by někdy měly být kombinovány s jiným systémem.

1.4 Možnost instalace

Možnosti instalace závisí převážně na stanoveném rozpočtu a vybraném systému. Důležitý faktor hraje také fakt, jestli se snažíme již postavenou nemovitost přebudovat na inteligentní dům, nebo jestli plánujeme inteligentní domácnost ještě před její stavbou. Mnohdy možnosti instalace limituje interiér či povaha již postaveného domu.

1.5 Spolehlivost a rizika

Spolehlivost inteligentních domů je sporné téma. Inteligentního dům bude tak spolehlivý, jak ho uživatel udělá. Při jeho tvorbě je nutno vzít v potaz mnoho různých faktorů, dát si pozor aby se některé funkce navzájem nerušily – například aby systém nespustil topení, zatímco jsou otevřená okna na vyvětrání. Systém je nutné ho navrhnout tak, aby se takovým omylům vyhnul. Nelze ho však udělat úplně dokonalým.

Jednou z hlavních hrozeb pro inteligentní domácnosti je napadení zvenčí a únik soukromých informací. Pokud se hacker nabourá do softwaru chytré domácnosti, může měnit její nastavení. Pokud je softwaru umožněno dálkové odemykání dveří, může dojít i k vyloupení domu.

Internetový server nalezeno.cz varuje: „Inteligentní systém zaznamenává **podrobná data mapující chod domácnosti**. Jen z přesných grafů spotřeby elektřiny je možné vyčíst, kdy není nikdo doma, různé zvyky, oblíbené televizní pořady apod. Pokud k tomu přičteme ještě živé videopřenosy z domu, potenciál pro špehování domácností a zneužití těchto informací je nedožrnný. “ [\[3\]](#)

Jedním z klíčových problémů je také možnost kolapsu či zničení řídicí (centrální) jednotky inteligentního domu. V takovém případě systém pravděpodobně přestane fungovat a s ním i všechny jeho funkce.

2 Praktická část

V praktické části se budu zabývat bytem se čtyřmi místnostmi (2+ kk), konkrétně koupelnou, ložnicí a obývacím pokojem s kuchyňským koutem a předsíní.

V první místnosti bytu, koupelně, se budu zabývat konkrétně ovládáním odsávání vzduchu na základě spínání světel, automatickém ohřívání vody na určitý čas a teplotu a zhasnutím světel po určitém čase bez pohybu v místnosti.

V druhé místnosti, pokoji, se pokusím zařídit, aby byla místnost konstantně vytápěná na určenou teplotu, aby se světla zhasla po 30 minutách bez pohybu v místnosti, nebo se ztlumila či vypnula v nastavený čas.

Ve třetí místnosti, obývacím pokoji s kuchyňským koutem nastavím, aby se digestoř zapnula v závislosti na zapnutí hořáků sporáků (pokud není zapnuta manuálně), dále budeme chtít, aby světla zhasla závisle na pohybu v místnosti. Také se budu zabývat vytápěním tak, aby šlo v programu nastavit na určitou teplotu.

V poslední místnosti, předsíni, budu chtít, aby se rozsvítila světla při pohybu v místnosti.

2.1 Popis výchozího stavu, definice cílového

Počítejme, že tepelná čidla, světla, vypínače, pohybová čidla atd. jsou rozmístěna po bytě podle plánu (viz foto1 a foto2). Tedy v koupelně světlo S1 a pohybové čidlo PIR1, odsávání a bojler spolu s teplotním čidlem T4.

V ložnici světlo S2, pohybové čidlo PIR2 a vypínač V2 a také tepelné čidlo T1 a elektrické podlahové vytápění TOP1.

V obývacím pokoji jsou umístěna světla S3.1, S3.2 a S3.3, dále k nim příslušné vypínače V3.1, V3.2 a V3.3 a pohybová čidla PIR3.1, PIR3.2 a PIR3.3 a dále tepelné čidlo T2 a T3 spolu s elektrickým podlahovým vytápěním TOP2.

V chodbě světlo S4, pohybové čidlo PIR4, vypínače V4.1, V4.2, V4.3 a vypínač VT spolu s řídicí jednotkou.

V cílovém stavu se bude světlo S1 zapínat a vypínat vypínačem V4.1, a nebo se automaticky vypne, pokud pohybové čidlo PIR1 nezaznamená pohyb po dobu 30 minut.

Světlo S2 se zapne sepnutím vypínače V2. Vypne se po sepnutí stejného vypínače, nebo pokud pohybové čidlo PIR2 nezaznamená žádný pohyb po dobu 30 minut.

Světlo S3.1 se zapne sepnutím vypínače V3.1 a vypne se sepnutím stejného vypínače nebo pokud pohybová čidla PIR3.1 a PIR3.2 nezaznamenají pohyb po dobu 40 minut.

Světlo S3.2 se zapne sepnutím vypínače V3.2 a stejný vypínač slouží i k jeho vypnutí. Pokud PIR3.1 a PIR3.2 nezaznamenají žádný pohyb po dobu 40 minut, světlo se vypne.

Světlo S3.3 se zapne sepnutím vypínače V3.3 a vypne se buď sepnutím toho samého vypínače nebo pokud pohybové čidlo PIR3.3 nezaznamená žádný pohyb po dobu 30 minut.

Světlo S4 se zapne změnou na vypínači V4.2 nebo V4.3 nebo když pohybové čidlo PIR4 zaznamená pohyb. Vypne se změnou na vypínači V4.2 nebo V4.3.

Odsávání v koupelně se zapne při rozsvícení světla S1 a vypne se automaticky po 5 minutách.

Podlahové vytápění TOP1 se zapne, pokud tepelné čidlo T1 zaznamená teplotu nižší než 21 °C. Vypne se po dosažení teploty 21 °C. Topení se dá trvale vypnout vypínačem VT.

Podlahové vytápění TOP2 se zapne, pokud tepelné čidlo T2 zaznamená teplotu nižší než 21 °C. Vypne se po dosažení teploty 21 °C. Topení se také dá trvale vypnout vypínačem VT.

Digestoř se zapne, pokud nebude zapnuta manuálně vypínačem VDI, když tepelné čidlo T3 zaznamená teplotu o minimálně 5 °C vyšší než čidlo T2. Vypne se buď vypínačem VDI, nebo pokud rozdíl teplot na čidlech T2 a T3 klesne pod 5 °C.

Vyhřívání bojleru zapne, když teplota vody zaznamenaná teplotním čidlem T4 klesne pod teplotu v časovém plánu a vypne se, pokud teplota vody překročí teplotu v časovém plánu.

2.2 Použité technologie

Cílem mé práce je vytvoření programu pro daný byt. Všechny ostatní prvky systému, s výjimkou řídicí jednotky AMiRiS99W3^[4] od značky Amit a s ním kompatibilního vývojového prostředí DetStudio, lze nahradit jinými modely.

Při realizaci používám jako pohybová čidla PIR1, PIR2, PIR3.1, PIR3.2, PIR3.3 a PIR4 detektor pohybu osob a rozbití skla JS-25 COMBO^[5] od firmy Jablotron. Tepelná čidla T1, T2 a T3 řeším použitím SENSIT snímače teploty modelu NS101^[6] z řady Tango. Tepelné čidlo do bojleru, tedy tepelné čidlo T4, realizuji použitím snímače teploty NS 131^[7] od firmy SENSIT. Jednopolové vypínače, tedy V2, V3.3, V4.1, V4.2 a V4.3, řeším použitím jednopolového vypínače z řady Tango^[8] od firmy ABB. Dvoupólový vypínač V3.1, V3. řeším použitím

dvoupólového vypínače řady Tango^[9] od firmy ABB.

Jak bylo již zmíněno, pro tvorbu programu nepotřebuji řešit model výstupů (ovládané objekty – např. světla, bojler), nýbrž jen relé, přes které je budu ovládat. Každé relé však musí být přizpůsobeno, aby vydrželo napětí ovládaného výstupu. Proto může být, např. pro světla a bojler, použito jiné relé. Pro bojler o výkonu 2 kW musí relé unést proud aspoň 8.7 A. Proto pro bojler, podlahové topení TOP1, TOP2 a odsávání s digestoří počítám s použitím relé VS116K^[10] od firmy ELKO EP. Pro zbytek výstupů je možné použít jiný typ relé, já však používám stejný model.

2.3 Tvorba programu

2.3.1 Výběr platformy, způsobu programování

Jako platformu použiji PLC Amit a programovací jazyk dle normy ČSN EN 61131-3. Jelikož program vytvářím pro zdroj od výrobce Amit, k jeho výrobě použiji program DetStudio, který je přímo určený pro zařízení Amit. DetStudio nabízí programování v strukturovaném textu a Ladder diagramu. Já použiji Ladder diagram, který je pro začátečníky jednodušší a rychlejší. Nejsou v něm však všechny funkce, které jsou ve strukturovaném textu, ty se tam však dají strukturovaným textem vložit. V mém případě mě tato skutečnost příliš neovlivňuje, jelikož můj program nebude mít žádné velmi pokročilé funkce.

Ladder diagram^[11] – grafický jazyk pro programování automatických systémů. Vzniká spojením jednoho či více prvků vstupů (v sériovém nebo paralelním zapojení) a výstupů.

2.3.2 Proměnné a Aliasy

Řídící automat cyklicky řeší vztahy mezi vstupy a výstupy. Vstupy a výstupy jsou proměnné, do nichž se načítají hodnoty fyzických vstupů (vstupní proměnné) a kterými se řídí výstupy (výstupní proměnné). Proměnné mohou být analogové (reálné číslo) nebo digitální (bit).

V tabulce je seznam proměnných se kterými pracuji.

Digitální vstupy		
PIR1	@s_pir_1	DI0 0
PIR2	@s_pir_2	DI0 1
V2	@s_v_2	DI0 2
PIR3.1	@s_pir_3_1	DI0 3
PIR3.2	@s_pir_3_2	DI0 4
PIR3.3	@s_pir_3_3	DI0 5
V3.1	@s_v_3_1	DI0 6
V3.2	@s_v_3_2	DI0 7
V3.3	@s_v_3_3	DI1 0
PIR4	@s_pir_4	DI1 1
V4.1	@s_v_4_1	DI1 2
V4.2	@s_v_4_2	DI1 3
V4.3	@s_v_4_3	DI1 4
VT	@s_v_t	DI1 5
VDI	@s_v_di	DI1 6

Analogové vstupy	
T1	t_t_1
T2	t_t_2
T3	t_t_3
T4	t_t_4

Digitální výstupy		
TOP1	@r_top_1	DO0 0
TOP2	@r_top_2	DO0 1
Odsávání	@r_vent_1	DO0 2
Digestoř	@r_vent_2	DO0 3
S1	@r_s_1	DO0 4
S2	@r_s_2	DO0 5
S3.1	@r_s_3_1	DO0 6
S3.2	@r_s_3_2	DO0 7
S3.3	@r_s_3_3	DO1 0
S4	@r_s_4	DO1 1
Bojler	@r_bo	DO1 2

Kdybych všechny tyto vstupy a výstupy zadala jako proměnné, seznam by byl nepřehledný a bylo by těžké v něm najít jednotlivé bity. Z tohoto důvodu jsem z digitálních vstupů a výstupů vytvořila tzv. aliasy, které se vyznačují právě znakem @ na jejich začátku. Tyto aliasy však musí být zařazené pod proměnnou. Jak je vidět v tabulce, jsou rozděleny do skupin DI (Digital Input) a DO (Digital Output). Dále jsou rozděleny do skupin DO0, DO1, DI0, DI1, protože řídicí jednotka, s kterou pracuji, má bity takto zařazené do skupin po osmi. Číslo v tabulce za skupinou udává, na jakém bitu proměnné je přesně daný bit umístěn.

V průběhu programování bude potřeba vytvářet pomocné proměnné a aliasy, ty však v této tabulce nejsou. Budou popsány dále v práci.

2.3.3 Rozdělení do procesů

Rozdělení podprocesů do jednotlivých procesů záleží čistě na periodě, při které potřebujeme, aby se podproces opakoval. Já jsem se rozhodla podprocesy rozdělit do dvou procesů a třetího procesu jen pro načtení vstupů a výstupů. První proces má periodu 200 ms s názvem *rychly_p* a druhý proces *topeni* má periodu 300 000 ms. V prvním procesu jsou zařazeny všechny podprocesy, které potřebují rychlé načtení – například ovládání světel a odsávání. Kdyby tyto podprocesy měly delší periodu, bylo by velké prodlení mezi změnou stavu na vstupu a mezi změnou stavu na výstupu. Například u ovládání světel by mezi sepnutím vypínače a rozsvícením světla byla nepříjemná prodleva. U druhého procesu, *topeni*, není potřeba periodu v řádu sekund. Pod proces patří podprocesy, u kterých není důležité jejich okamžité načtení – například vytápění.

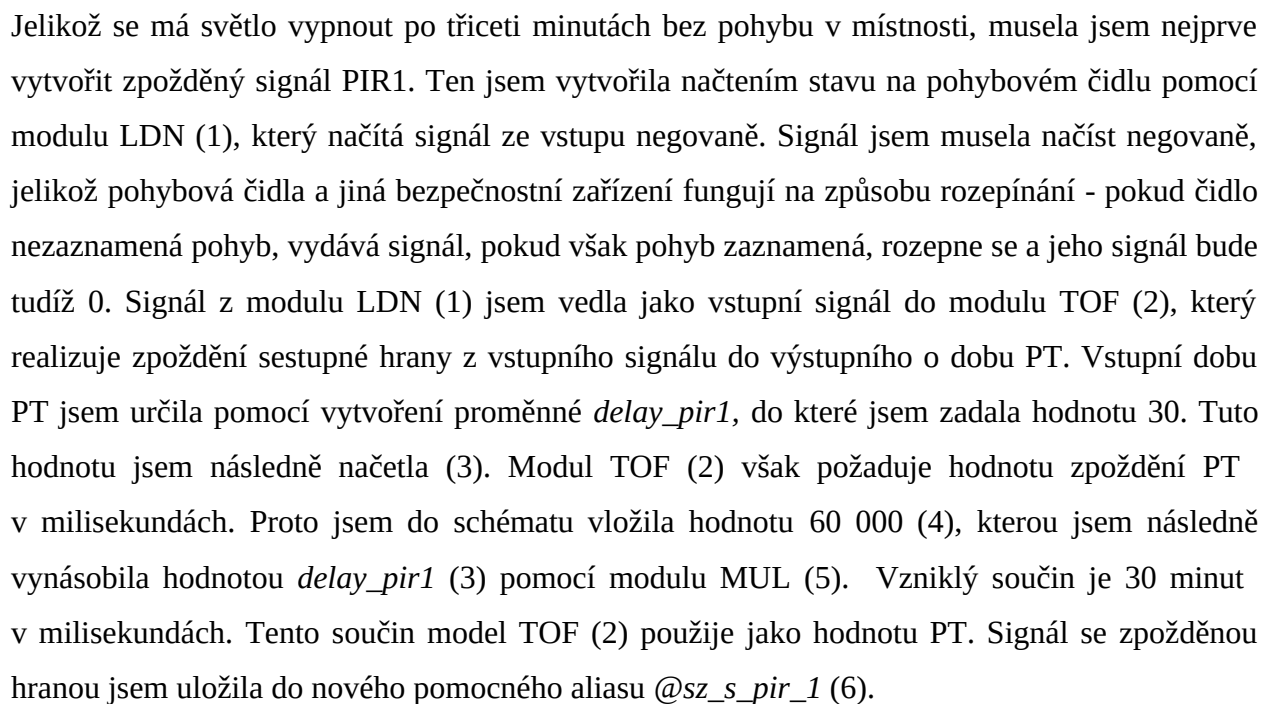
2.3.4 Tvorba jednotlivých procesů

V dalším textu se budu věnovat tvorbě jednotlivých procesů a jejich podprocesů. K vysvětlení přesné tvorby programu budu, pro lepší vizualizaci, používat i výřezy ze samotného programu.

2.3.4.1 rychly_p

Jak již bylo zmíněno, jedná se o proces s periodou 200 ms. Z toho důvodu do něj spadá zejména ovládání světel. Konkrétně ovládání S1, S2, S3.1, S3.2, S3.3, S4, digestoře a odsávání v koupelně.

“V cílovém stavu bude světlo S1 zapnuto vypínačem V4.1. Vypínat se bude též vypínačem V4.1 a nebo se automaticky vypne pokud pohybové čidlo PIR1 nezaznamená pohyb po dobu 30 minut.“



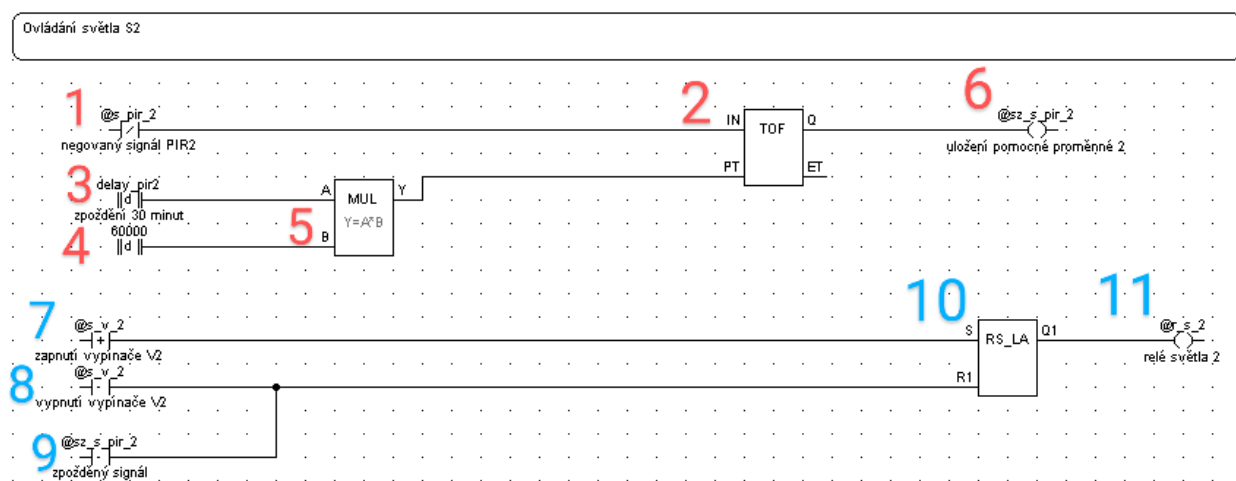
16

do druhého dochází skokově).

Vypnutí vypínače jsem řešila použitím modulu LD_F do kterého jsem načetla signál @s_v_4_1. Modul LD_F je opak modulu LD_R, tudíž zaznamenává pouze načtení sestupné hrany (vypnutí). S tímto modulem jsem paralelně zapojila s další modul LD_F, tentokrát však s hodnotou zpožděného signálu @sz_s_pir_1 (9). Tento signál jsem pak vedla do vstupu R1 modulu RS_LA (10). Signál z výstupu tohoto modulu jsem uložila do aliasu signálu relé světla @r_s_1 (11).

Ovládání světla S2

“Světlo S2 se zapne sepnutím vypínače V2. Vypne se po sepnutí stejného vypínače, či pokud pohybové čidlo PIR2 nezaznamená žádný pohyb po dobu 30 minut.”



Jelikož je definice cílového stavu velmi podobná jako cílový stav u světla S1, postup tvorby programu ovládání bude také velmi podobný.

Stejně jako u světla S1, musela jsem nejprve vytvořit zpožděný signál PIR2. Ten jsem vytvořila stejným způsobem jako zpožděný signál PIR1, jenom s jinými hodnotami a signály.

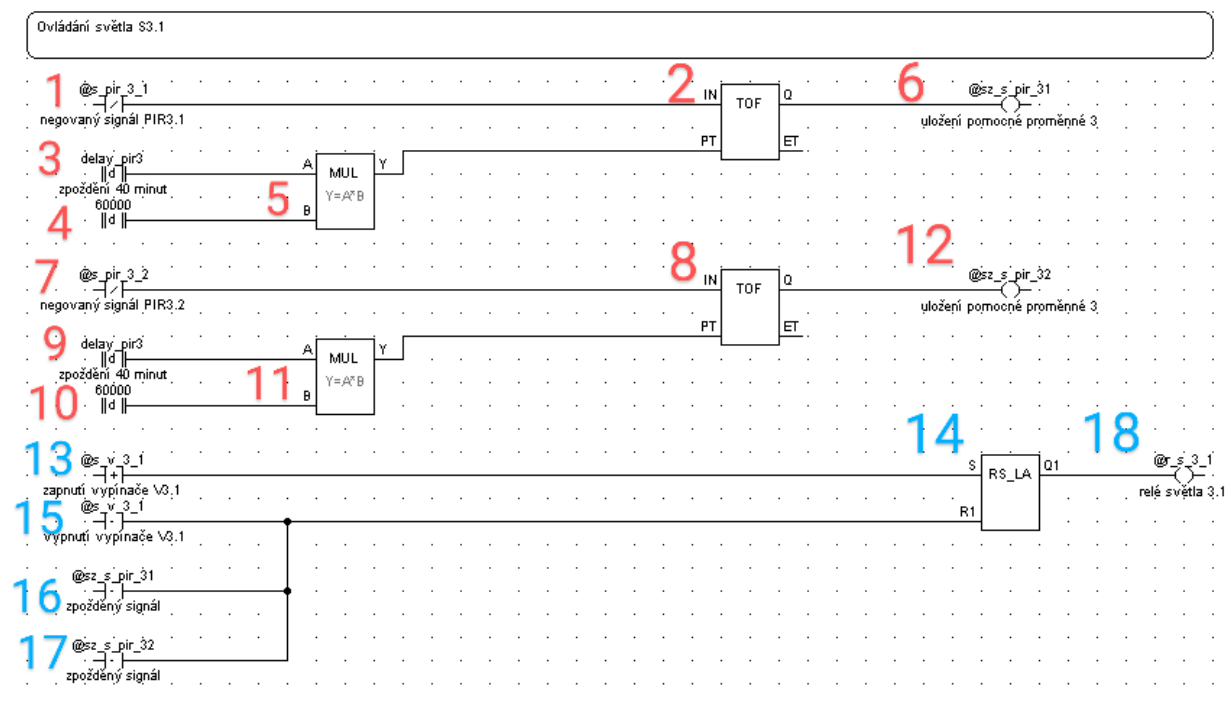
Načetla jsem stav na pohybovém čidle PIR2 (1), tento signál jsem dál vedla do modulu TOF (2). Vytvořila jsem pomocnou proměnnou *delay_pir2* (3), které jsem nastavila hodnotu 30. Tuto hodnotu jsem následně pomocí modelu MUL (5) vynásobila hodnotou 60 000 (4). Tento signál jsem použila jako PT pro modul TOF (2). Výsledný signál jsem uložila do pomocného aliasu zpožděného signálu @sz_s_pir_2 (6).

U samotné tvorby jsem opět pro zapnutí použila modul LD_R, do kterého jsem načetla signál

@s_v_2 (7). Tento signál jsem dále vedla do modelu RS_LA (10). Pro vypnutí jsem jako u předchozího světla použila model LD_F, do kterého jsem vložila signál @s_v_2 (8). Spolu s ním jsem paralelně zapojila zpožděný signál pohybového čidla @sz_s_pir_2 (9). Tento signál jsem následně vedla do modelu RS_LA (10), z kterého jsem výsledný signál uložila do signálu relé světla @r_s_2 (11).

Ovládání světla S3.1

“Světlo S3.1 se zapne sepnutím vypínače V3.1 a vypne se sepnutím stejného vypínače, nebo pokud pohybová čidla PIR3.1 a PIR3.2 nezaznamenají pohyb po dobu 40 minut.”



Postup tvorby programu je stejný jako u předešlých světél, pouze s přidáním dalšího pohybového čidla. Pro vytvoření zpožděného signálu jsem paralelně zapojila modul LDN s načtením signálu @s_pir_3_1 (1). Tento signál jsem následně zavedla do modelu TOF (3). Vytvořila jsem pomocnou proměnnou delay_pir3 (4) a do ní jsem vložila hodnotu 40, kterou jsem následně pomocí modulu MUL (5) vynásobila hodnotou 60 000 (5). Výslednou hodnotu jsem použila jako PT pro modul TOF (3). Výsledný signál jsem uložila do aliasu zpožděného signálu @sz_s_pir_31 (7).

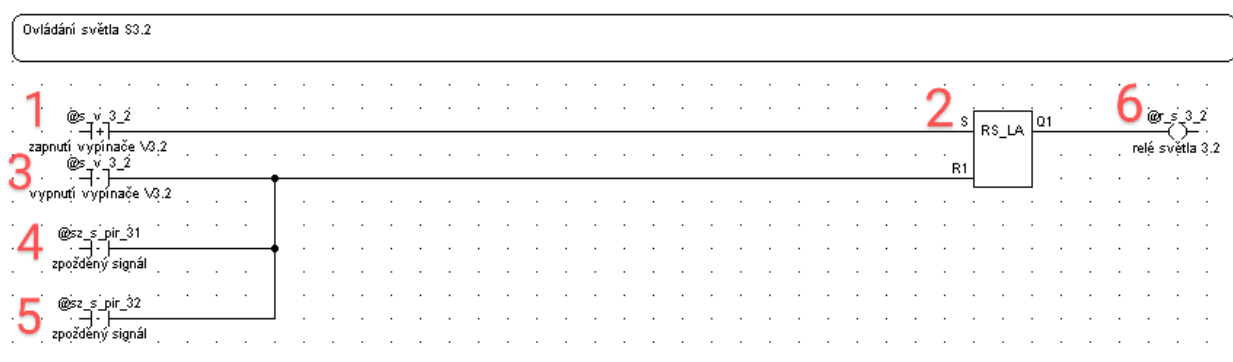
Pro vytvoření zpožděného signálu jsem vytvořila stejný proces, pouze jsem načítala stav `@s_pir_3_2` (7) a výsledný signál jsem uložila do proměnné `@sz_s_pir_32` (12). Tvořila jsem dvě různé proměnné aby když jedno čidlo vyplo či bylo rozbito, druhé mohlo stále ovládat světlo.

Pro zapnutí jsem do modulu LD_R načetla signál `@s_v_3_1` (13), který jsem následně zavedla do modulu RS_LA (14).

Pro vypnutí jsem paralelně zapojila signál `@s_v_3_1` v modulu LD_F (15), zpožděný signál `@sz_s_pir_31` (16) a `@sz_s_pir_32` (17). Výsledný signál jsem vedla do modulu RS_LA (14) a signál z něj jsem uložila do relé světla `@r_s_3_1` (18).

Ovládání světla S3.2

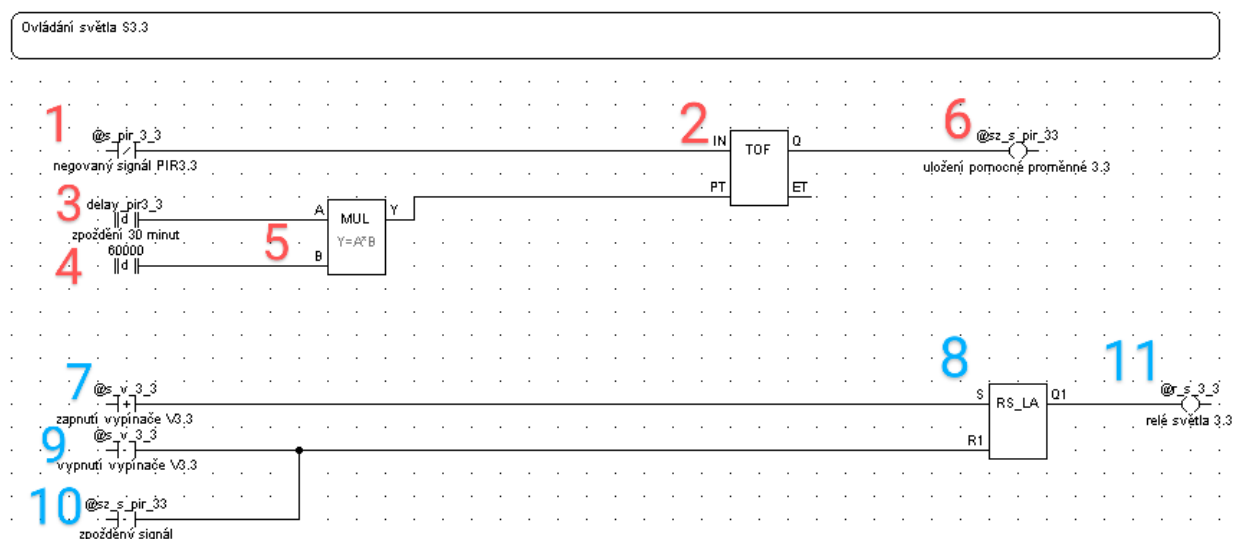
“Světlo S3.2 se zapne sepnutím vypínače V3.2 a stejný vypínač slouží i k jeho vypnutí. Pokud PIR3.1 a PIR3.2 nezaznamenají žádný pohyb po dobu 40 minut, světlo se vypne.”



Protože ovládání tohoto světla používá stejná pohybová čidla jako světlo S3.1, jako zpožděné signály pohybových čidel použiji signál `@sz_s_pir_31` (4) a `@sz_s_pir_32` (5), které jsem vytvořila při tvorbě ovládání S3.1. Postup tvorby procesu je také stejný. Modul LD_R se signálem `@s_v_3_2` (1) jsem napojila na modul RS_LA (2). Modul LD_F se signálem `@s_v_3_2` (3) jsem zapojila paralelně s zpožděným signálem `@sz_s_pir_31` (4) a `@sz_s_pir_32` (5). Výsledný signál jsem následně vedla na vstup R1 modulu RS_LA (2), jehož výsledný signál jsem uložila do relé `@r_s_3_2` (6).

Ovládání světla S3.3

“Světlo S3.3 se zapne sepnutím vypínače V3.3 a vypne se buď sepnutím toho samého vypínače nebo pokud pohybové čidlo PIR3.3 nezaznamená žádný pohyb po dobu 30 minut.”

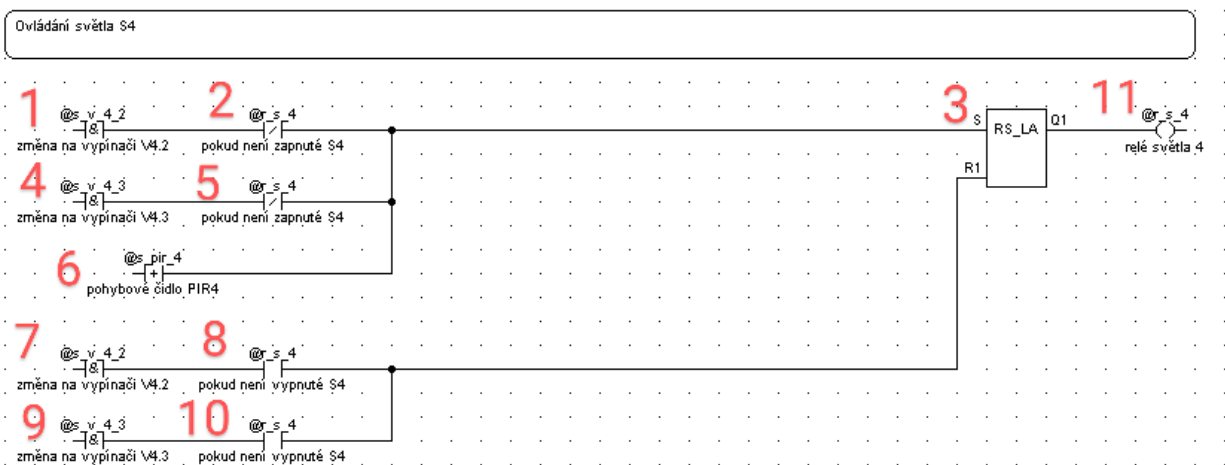


Jako u všech předešlých případů bylo napřed nutno vytvořit zpožděný signál pohybového čidla. Načetla jsem stav na pohybovém čidle PIR3.3 pomocí LDN modulu s vloženým signálem `@s_pir_3_3` (1), tento signál jsem následně zavedla do modulu TOF (2). Jako druhý vstup do tohoto modulu jsem použila proměnnou `delay_pir3_3` (3) s hodnotou 30 vynásobenou pomocí modulu MUL (5) o hodnotu 60 000 (4). Výsledný signál z modulu TOF (2) jsem uložila do pomocného aliasu `@sz_s_pir_33` (6).

Zapnutí jsem řešila zapojením modulu LD_R se signálem `@s_v_3_3` (7) do modulu RS_LA (10). Vypnutí paralelním zapojením modulu LD_F se signálem `@s_v_3_3` (8) spolu se zpožděným signálem pohybového čidla `@sz_s_pir_33` (9). Jejich výsledný signál jsem použila jako druhou vstupní hodnotu pro modul RS_LA (10), z kterého jsem výsledný signál uložila do signálu relé světla `@r_s_3_3` (11).

Ovládání světla S4

“Světlo S4 se zapne změnou na vypínači V4.2 nebo V4.3 a když pohybové čidlo PIR4 zaznamená pohyb. Vypne se změnou na vypínači V4.2 nebo V4.3.”

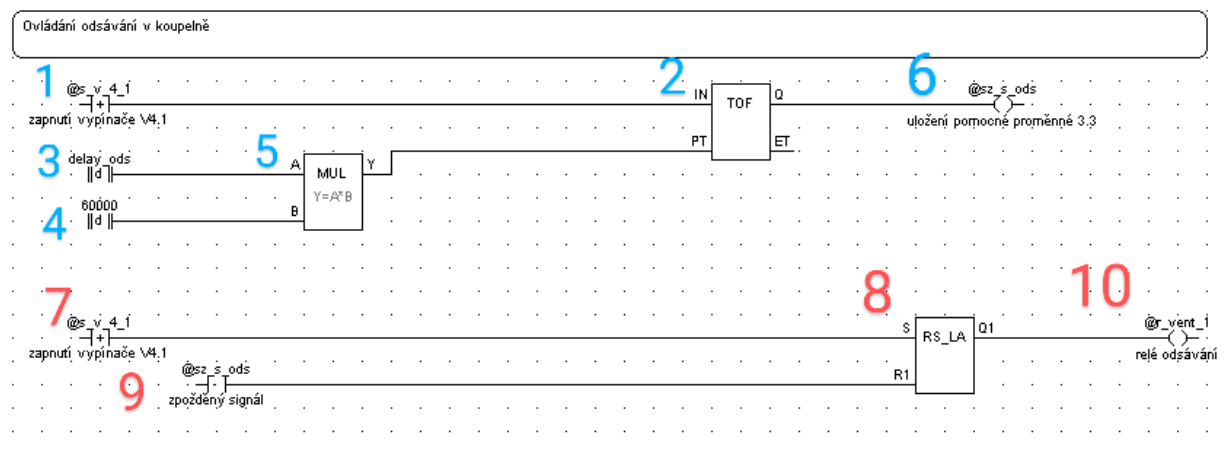


Jelikož zadání pro světlo S4 je jiné než zadání ostatních světel, bylo nutné použít jiné moduly. Zapnutí světla má být ovládáno změnou na jednom ze dvou vypínačů či pohybovým čidlem. Aby ho bylo možné zapnout pokud se na vypínači odehraje jakákoliv změna, ne pouze když je vypínač v poloze zapnutí je nutné použití modulu LD_RF. Ten načítá pulz podle náběžné i sestupné hrany - reaguje na každou změnu. Do tohoto modulu jsem načetla signál vypínače V4.2 @s_v_4_2 (1). Jenže pro zapnutí potřebuji aby tento signál prošel pouze pokud světlo již není zapnuté. To jsem řešila použitím modulu LDN, který načte daný bit negovaně. Do tohoto modulu jsem vložila signál relé světla @r_s_4 (2). Na stejném principu jsem vytvořila i zapínání vypínačem V4.3. Tedy do modulu LD_RF jsem vložila signál @s_v_4_3 (4), který jsem sériově zapojila s modulem LDN s signálem @r_s_4 (5). Následně jsem potom s oběma těmito rameny paralelně zapojila modul LD_R se signálem pohybového čidla @s_pir_4 (6). Výsledný signál jsem vedla do modulu RS_LA (3).

Pro vypnutí jsem použila stejný princip. Sériově jsem zapojila modul LD_RF s signálem @s_v_4_2 (7) s modulem LD, který načte daný bit, a vložila jsem do něj signál relé světla @r_s_4 (8). Stejně jsem zapojila modul LD_RF se signálem @s_v_4_3 (9) a modul LD se signálem @r_s_4 (10). Tyto dvě ramena jsem spolu následně zapojila paralelně a jejich signál jsem vedla jako druhý vstup do modelu TOF (3). Výsledný signál tohoto modulu jsem uložila do signálu relé světla S4 @r_s_4 (11).

Ovládání odsávání

“Odsávání v koupelně se zapne při rozsvícení světla S1 a vypne se automaticky po 5 minutách.”



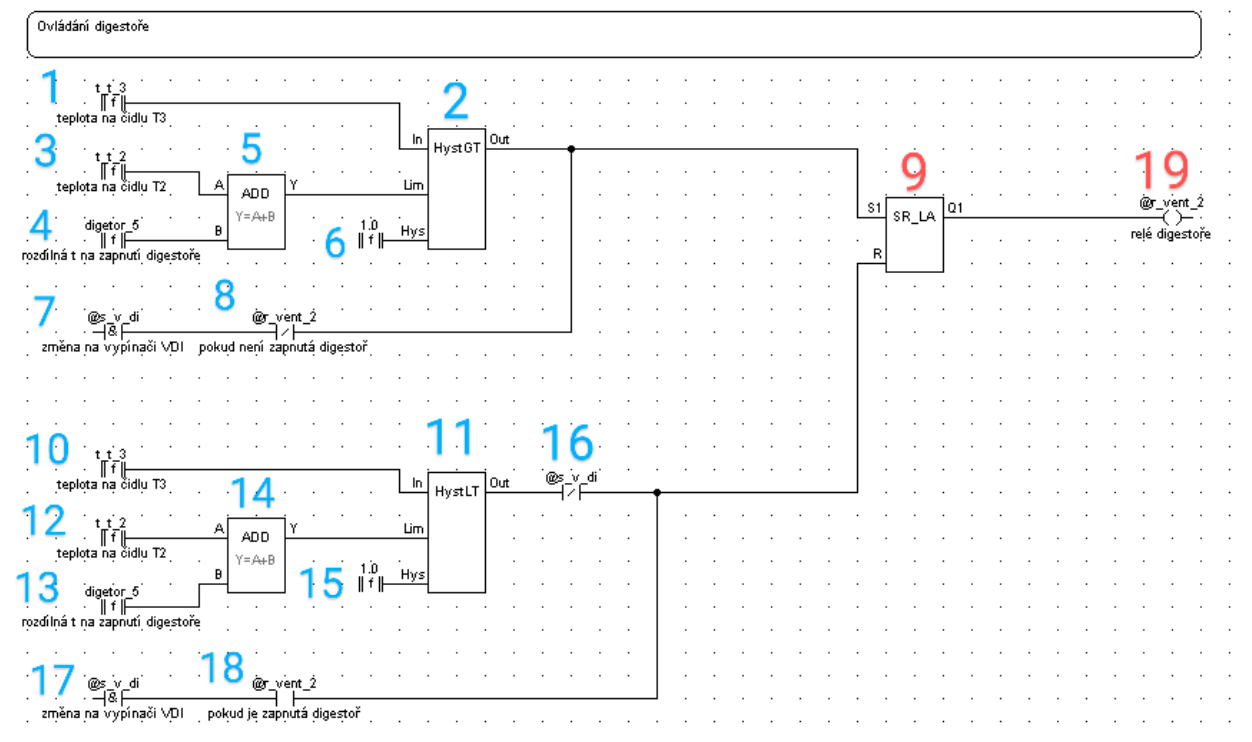
Podobně jako u ovládání většiny světel jsem nejprve musela vytvořit zpožděný signál vstupu, v tomto případě o pět minut zpožděný signál vypínače V4.1. Ten jsem vytvořila zapojením modulu LD_R s aliasem vypínače V4.1 @s_v_4_1 (1) do modulu TOF (2). Hodnotu PT modulu TOF (2) jsem vytvořila vynásobením pomocné proměnné *delay_ods* (3), které jsem nastavila hodnotu 5, pomocí modulu MUL (5) o hodnotu 60 000 (4). Výsledný signál jsem uložila do pomocného aliasu @sz_s_ods (6).

Pro zapnutí jsem pomocí modulu LD_R s signálem @s_v_4_1 (7) načetla zapnutí vypínače V4.1 a tento signál jsem vedla do modulu RS_LA (8).

Pro vypnutí jsem pomocí modulu LD_F načetla zpožděný signál V4.1 @sz_s_ods (9) a zavedla jsem ho do modulu RS_LA (8). Výsledný signál z tohoto modulu jsem uložila do signálu relé @r_vent_1 (10).

Ovládání digestoře

“Digestoř se zapne, pokud nebude zapnuta manuálně vypínačem VDI, když tepelné čidlo T3 zaznamená teplotu o 5°C vyšší než čidla T2. Vypne se buď vypínačem VDI a nebo pokud teplota na čidlu T2 není o 5°C nižší než čidla T3.”



Ovládání digestoře je jedna z nejsložitějších částí celého programu. Jelikož je v zadání ovládání tepelným čidlem, což je analogový vstup, bylo nutné používat jiné moduly než pro digitální vstupy. Pomocí modulu LDf, který načítá hodnotu typu float, jsem načetla teplotu na čidlu T3 t_t_3 (1). Signál z tohoto modulu jsem následně vedla do modulu HystGT (2) jako vstupní hodnotu *In*. Modul HystGT (2) ze svých tří vstupních hodnot *In*, *Lim*, *Hyst* porovná zda je hodnota $In > Lim$ s přihlédnutím k nastavené hysterezi *Hyst*. Dále jsem pomocí modulu LDf načetla teplotu na čidle T2 t_t_2 (3) a tu jsem zadala jako vstupní hodnotu A pro modul ADD (5). Modul ADD (5) provede aritmetický součet ze svých vstupů A a B. Jako hodnotu B jsem do tohoto modulu zavedla signál z modulu LDf s hodnotou pomocné proměnné *digetor_5* (4), která má hodnotu 5 °C. Výsledný signál jsem použila jako vstupní hodnotu *Lim* pro modul HystGT (2). Jako hodnotu *Hyst* jsem použila natvrdo nastavenou hodnotu 1 (6) pomocí modulu

LDf. Sériově jsem zapojila modul LD_RF s načtením aliasu vypínače VDI @s_v_di (6) a modul LDN s načtením stavu relé digestoře @r_vent_2 (7). Modul LDN načítá negovaný bit. Výsledný signál jsem paralelně zapojila s signálem vycházejícím z HystGT (2). Jejich výsledný signál jsem vedla do modulu SR_LA (9). Modul SR_LA (9) je RS klopný obvod s dominantním set. Jeho použití v tomto procesu mi zajistí že digestoř bude moct být zapnuta, aniž by ji rovnou vyplo tepelné čidlo.

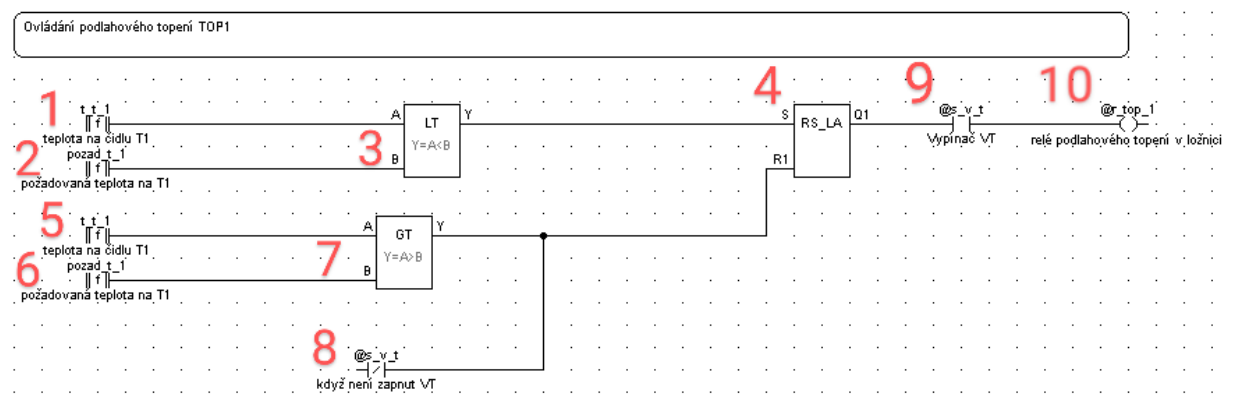
Pro vypnutí jsem načetla t_t_3 (10) pomocí modulu LDf a tuto hodnotu jsem použil jako vstup *In* pro modul HystLT (11). Modul HystLT (11) je opakem modulu HystGT (2), tedy ze svých tří vstupních hodnot *In*, *Lim*, *Hyst* porovná zda je hodnota $In < Lim$ s přihlédnutím k nastavené hysterezi *Hyst*. Jako hodnotu *Lim* jsem do tohoto modulu vedla výslednou hodnotu z modulu ADD (14), který aritmeticky sečetl hodnotu z vstupu A, tedy hodnotu t_t_2 (12), a hodnotu ze vstupu B, tedy hodnotu *digetor_5* (4). Jako hodnotu *Hyst* jsem stejně jako u modulu HystGT (2) použila 1 (15). Signál z modulu HystLT (11) jsem vedla do modulu LDN s signálem vypínače @s_v_di (16). Následně jsem sériově zapojila modul LD_RF s aliasem vypínače VDI @s_v_di (17) s modul LD s načtením stavu relé @r_vent_2 (18). Signál z tohoto zapojení jsem paralelně zapojila s signálem z modulu LDN @s_v_di (16) a jejich výsledný signál jsem zavedla do modulu SR_LA (9). Signál z tohoto modulu jsem uložila do signálu relé digestoře @r_vent_2 (19).

2.3.4.2 Topení

Jedná se o proces s periodou 300 000 ms a spadá pod něj ovládání podlahových topení TOP1, TOP2 a boileru.

Ovládání podlahového topení TOP1

“Podlahové vytápění TOP1 se zapne, pokud tepelné čidlo T1 zaznamená teplotu nižší než 21°C. Vypne se po dosažení teploty 21°C. Topení se dá trvale vypnout vypínačem VT.”

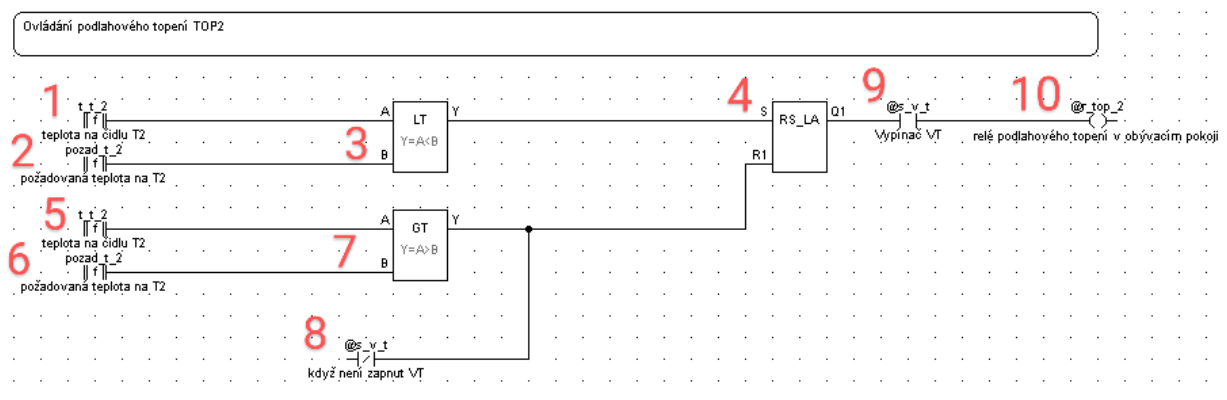


Nejprve jsem pomocí modulu LDf načetla hodnotu t_t_1 (1), kterou jsem následně pomocí modulu LT (3), který porovnává jestli vstup $A < B$, porovnávala s hodnotou pomocné proměnné $požad_t_1$ (2). Této proměnné jsem nastavila hodnotu požadované teploty v pokoji, tedy 21 °C. Signál z modulu LT (3) jsem vedla do modulu RS_LA (4).

Pro vypnutí jsem také načetla hodnotu t_t_1 (5) a porovnávala jsem ji s hodnotou $požad_t_1$ (6) pomocí modulu GT (7), který porovnává zda vstup $A > B$. Signál z modulu GT (7) jsem paralelně zapojila s modulem LDN s signálem $@s_v_t$ (8). Tímto jsem zajistila že se topení vypne, pokud bude vypnut vypínač VT. Výsledný signál jsem vedla do modulu RS_LA (5), signál z tohoto modulu sériově zapojila spolu s modulem LD s signálem vypínače $@s_v_t$ (9). Tímto jsem zajistila že topení nebude topit když bude vypínač vypnutý. Signál z tohoto modulu jsem následně uložila do signálu relé topení $@r_top_1$ (6).

Ovládání podlahového topení TOP2

“Podlahové vytápění TOP2 se zapne, pokud tepelné čidlo T2 zaznamená teplotu nižší než 21°C. Vypne se po dosažení teploty 21°C. Topení se také dá trvale vypnout vypínačem VT.”

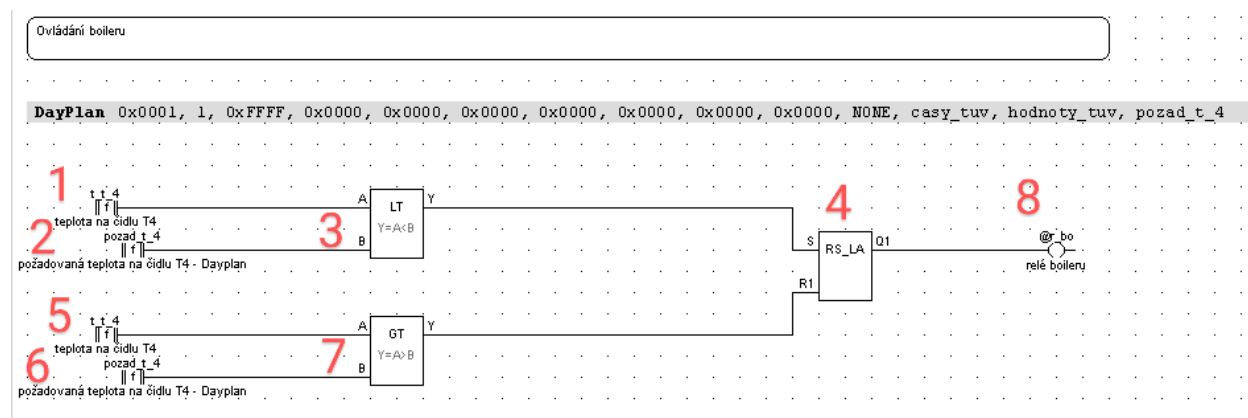


Ovládání podlahového topení TOP2 je až na tepelné čidlo a proměnnou požadovenné teploty stejné jako ovládání TOP1. To znamená že jsem pomocí modulu LT (3) porovnávala hodnotu t_{t_2} (1) a $pozad_t_2$ (2) a signál z tohoto modulu jsem vedla do modulu RS_LA (4).

Pro vypnutí jsem porovnávala hodnotu t_{t_2} (5) s hodnotou $pozad_t_2$ (6) pomocí modulu GT (7). Tento modul jsem paralelně zapojila s modulem LDN se signálem @s_v_t (8). Jejich výsledný signál jsem vedla do modulu RS_LA (4), který jsem sériově zapojila s modulem LD se signálem @s_v_t (9). Výsledný signál jsem uložila do signálu relé podlahového topení @r_top_2 (10).

Ovládání boileru

“Vyhřívání boileru zapne když teplota vody zaznamenaná teplotním čidlem T4 klesne pod teplotu v časovém plánu a vypne se pokud teplota vody překročí teplotu v časovém plánu.



Jelikož je v zadání programu ovládání teploty pomocí časového plánu, nejprve jsem se zabýval jím. Jelikož modul pro časový plán pro ladder diagram není, musela jsem do programu vložit modul strukturovaného textu - DayPlan. Modul **DayPlan** provádí plánování hodnoty do databázové proměnné *Output* z matice hodnot *Values* a časového plánu *Times*. Jde do něj nastavit speciální plán pro každý den týdnu a výjimky pro svátky. To však pro mé účely není potřeba. DayPlan pro mou práci se skládá z jednoho plánu pro všechny dny v týdnu. Jako hodnotu *Values* jsem do něj vložila proměnnou *hodnoty_tuv*, která má v sobě šest hodnot pro různé časy dne (v mém případě vždy 55°C). Za hodnotu *Times* jsem dosadila pomocnou proměnnou *casy_tuv*, která má šest hodnot určující čas v sekundách. Přesněji časy 0:00, 4:00, 8:00, 12:00, 16:00, 20:00. Jako *Output* jsem nastavila pomocnou proměnnou *pozad_t_4*.

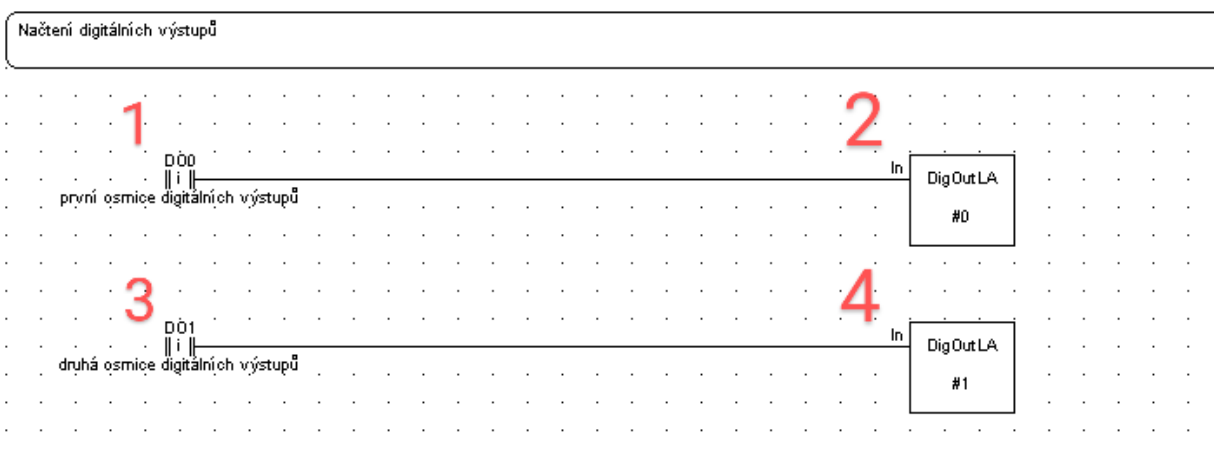
Samotný program jsem tvořila načtením teplot na čidlu T4 pomocí modulu LDf s proměnnou *t_t_4* (1) a porovnáním této proměnné se požadovanou teplotou *pozad_t_4* (2) pomocí modulu LT (3). Signál z tohoto modulu jsem dále vedla do modulu RS_LA (4).

Pro vypnutí jsem načetla hodnotu *t_t_4* (5) a pomocí modulu GT (7) ji porovnála s požadovanou hodnotou *pozad_t_4* (6). Výsledný signál jsem vedla do modulu RS_LA (4). Signál z tohoto modulu jsem uložila do signálu relé boileru *@r_bo* (8).

2.3.4.3 vstupy_vy

Proces s periodou 100 ms. Obsahuje načtení analogových vstupů (T1, T2, T3, T4), digitálních vstupů a digitálních výstupů.

Načtení digitálních výstupů

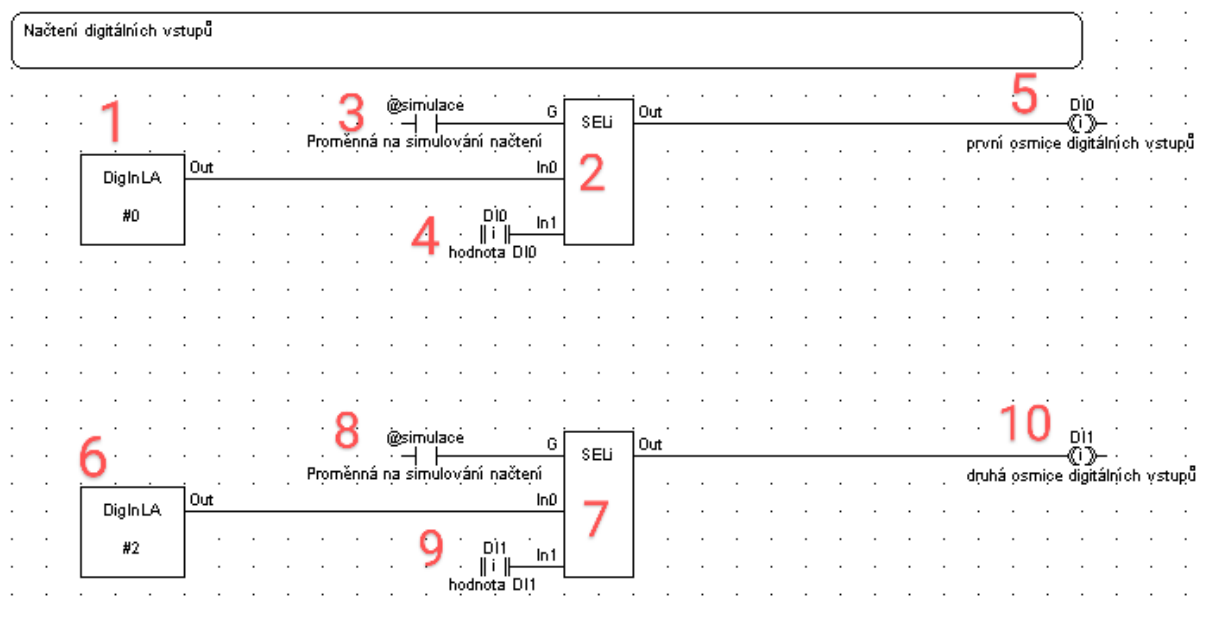


Načtení digitálních vstupů jsem uskutečnila pomocí použití modulu DigOutLA (2 a 4) a jeho následným propojením s LDi (1 a 3). Modul LDi načítá hodnotu int, což je v mém případě proměnná s aliasy. Modul DigOutLA vstupní hodnotu int zapíše jako šestnáctici digitálních výstupů do kanálu DO (Digital Output).

Pro načtení proměnné DO0 jsem do modulu LDi (1) zadala hodnotu DO0 a vedla jsem signál do modulu DigOutLA, kterému jsem nastavila aby hodnotu zapsal do kanálu DO0 (2).

Pro načtení proměnné DO1 jsem do modulu LDi (3) zadala hodnotu DO1 a vedla jsem signál do modulu DigOutLA, kterému jsem nastavila aby hodnotu zapsal do kanálu DO1 (4).

Načtení digitálních vstupů

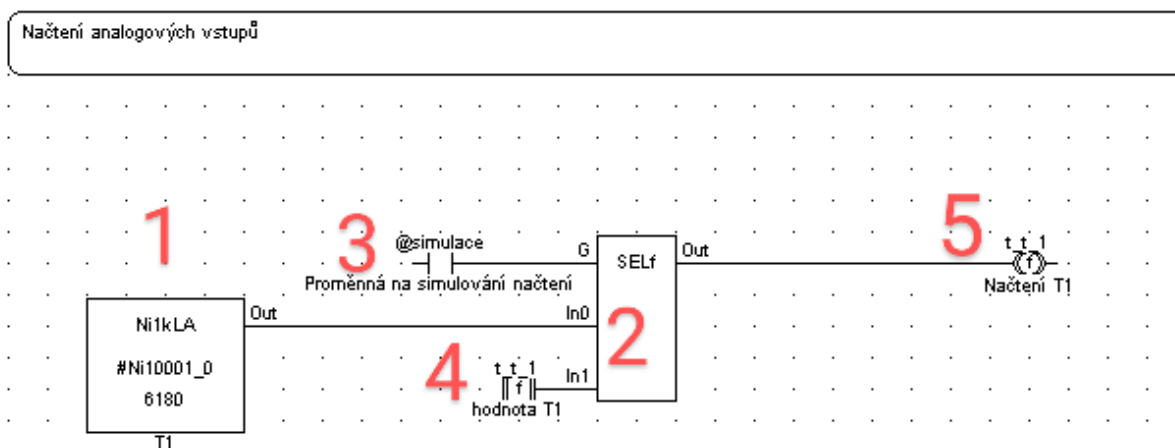


Abych mohla spustit program bez nutnosti zapojit veškeré vstupy se kterými pracuje, vytvořila jsem jeho simulaci. Tu jsem udělala pomocí modulu SELi (2 a 7), který vybere jednu z hodnot int (*In0*, *In1*) na základě řídicího vstupu *G*. Jako hodnotu *In0* jsem tomuto modulu nastavila signál z modulu DigInLA (1 a 6). Modul DigInLA čte šestnáct digitálních vstupních signálů DI (Digital Input) a hodnotu uloží do výstupu *Out*. Jako hodnotu *In1* jsem pro modul SELi použila hodnotu *DI_n*, tudíž když modul udělá simulované načtení s hodnotou *In1*, načte pouze již stanovenou hodnotu. Jako vstupní hodnotu *G* jsem použila pomocný alias @simulace (3 a 8).

Při načtení prvních šestnácti vstupů jsem modulu DigInLA nastavila kanál DI0 (1) a za vstupní hodnotu *In1* pro modul SELi (2) proměnnou DI0 (4). Výsledný signál jsem uložila pomocí modulu STi (ukládá hodnotu int do proměnné) do proměnné DI0.

Načtení dalších šestnáct čísel jsem udělala stejně, pouze jsem kanály a proměnnou zaměnila za DI1.

Načtení analogových vstupů



Pro načtení analogových vstupů jsem také vytvořila jeho simulaci. Tu jsem udělala pomocí modulu SELf (2), který je identický jako modul SELi, pouze pracuje s hodnotami float na rozdíl s int. Jako vstup *In0* jsem použila signál z modulu Ni1kLA (1), který čte hodnotu z odporového snímače Ni1000, kterému jsem dala hodnotu T1. Jako vstup *In1* jsem použila proměnnou *t_t_1* (4) a jako vstup *G* proměnnou *@simulace* (3). Výsledný signál jsem pomocí modulu STf (uloží hodnotu float do proměnné) uložila do proměnné *t_t_1* (5).

Načtení T2, T3, T4 jsem provedla stejně, pouze jsem nahradila hodnoty T1 o hodnotu příslušného T.

2.4 Ladění programu

Pro ladění a zkoušku programu jsem sestrojila model vstupů a výstupů v bytě. Od každého typu vstupu jsem zapojila jeden do obvodu obsahujícího řídicí jednotku. Tedy 1 pohybové čidlo, 1 vypínač, 1 teplotní čidlo, teplotní čidlo boileru. Pro výstupy jsem do obvodu zapojila 1 relé. Pro lepší vizualizaci modelu (viz. foto3). Samotné ladění programu spočívalo v generaci programu a jeho nahrání do řídicí jednotky. Následně jsem v DetStudios zapla režim *Watch*, který mi dovoľoval dívat se na proces a jeho průběh. V režimu *Watch* je možné vidět hodnoty všech modulů, popřípadě jak dlouho proces trvá, či jak přesně probíhá. Na obrázku je proces ovládání digestoře v režimu *Watch* (viz. foto4). Postupně jsem zapojila obvod každého procesu a vyzkoušela jsem jejich funkčnost a průběh. Pro rychlejší odzkoušení jsem všem pomocným proměnnám *delay* dočasně nastavila hodnotu 1. Během ladění jsem narazila na problém, a to, že mé pohybové čidlo snímalo pohyb až moc citlivě. Pro snazší a rychlejší vyzkoušení programu

jsem tedy pohybové čidlo zablokovala alobalem, který jsem sejmula pouze pokud jsem potřebovala by čidlo pohyb snímalo.

Zjistila jsem že procesy fungují tak jak je jim určeno, kromě procesu ovládání světla S4. Tento proces obsahuje problém v zapojení a ovládání v pohybovém čidlu PIR4. Toto čidlo má zapnout světlo 4, pokud zaznamená pohyb. Problém spočívá v vypínání světla. Oba dva vypínače od tohoto světla se nacházejí ve stejné místnosti jako pohybové čidlo. Pokud tedy někdo bude světlo chtít vypnout, musí být v místnosti. Tím pádem ho zaznamená pohybové čidlo a vypnuté světlo okamžitě zase zapne.

Řešení tohoto problému by pravděpodobně bylo nastavit čidlu PIR4 prodlevu několik sekund po každém zaznamenaném pohybu. Tudíž by šlo světlo vypnout bez toho aby se samo zpět zapnulo. Toto řešení je však momentálně mimo mé znalosti, proto ho plánuji řešit příští rok.

3 Závěr

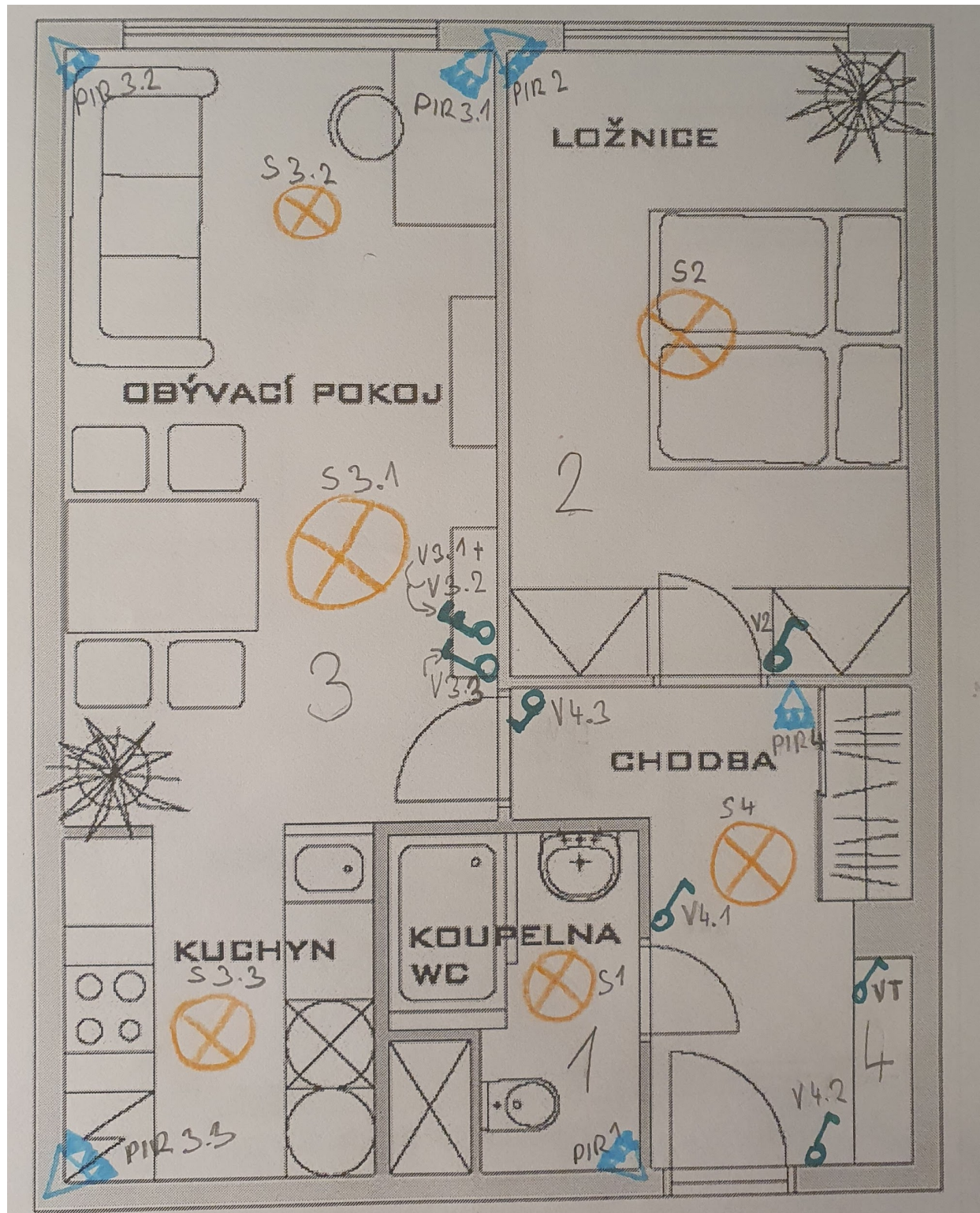
Vypracování této práce pro mě byla nová, ale vítaná zkušenost. Téma práce se mi líbilo zpracovávat a psaní programu pro mě byla cenná zkušenost. Naučit se tvořit program v DetStudiosu pro mě bylo velmi přínosné a nabyté zkušenosti jistě v budoucnu použiji. Porozumět všem funkcím a správný výběr modulů jsem se učila převážně z nápovědy samotného DetStudia a získala jsem mnoho znalostí, z nichž jen zlomek je použit v této práci. Nejvíce přínosná část práce pro mě bylo určitě ladění programu. Možnost vidět jak program pracuje v reálném čase a ovládat ho pomocí fyzických vstupů mi mnoho přinesla. Samotný program který jsem vytvořila může být (kromě ovládání světla S4) použitý pro účely ovládání bytu 2+kk podobnému jako ten, s kterým jsem pracovala já. Program je popsán a lze z něj lehce vyčíst, co se v které části odehrává, tudíž by případně bylo lehké ho upravit pro potřeby uživatele.

Příští školní rok plánuji na tuto práci navázat. Chtěla bych vyřešit problém se světlem S4 a přidat nové procesy. Tyto procesy by řešily již pokročilejší funkce. Například zabezpečení bytu, klimatizaci, nebo vylepšení stávajících funkcí. Chtěla bych pro program vytvořit vizualizaci a umožnit uživateli ovládat program z mobilního telefonu. Také bych chtěla uživateli umožnit aby mohl světla ztlumit na určité procento jasů, jak jsem původně zamýšlela u světla S2.

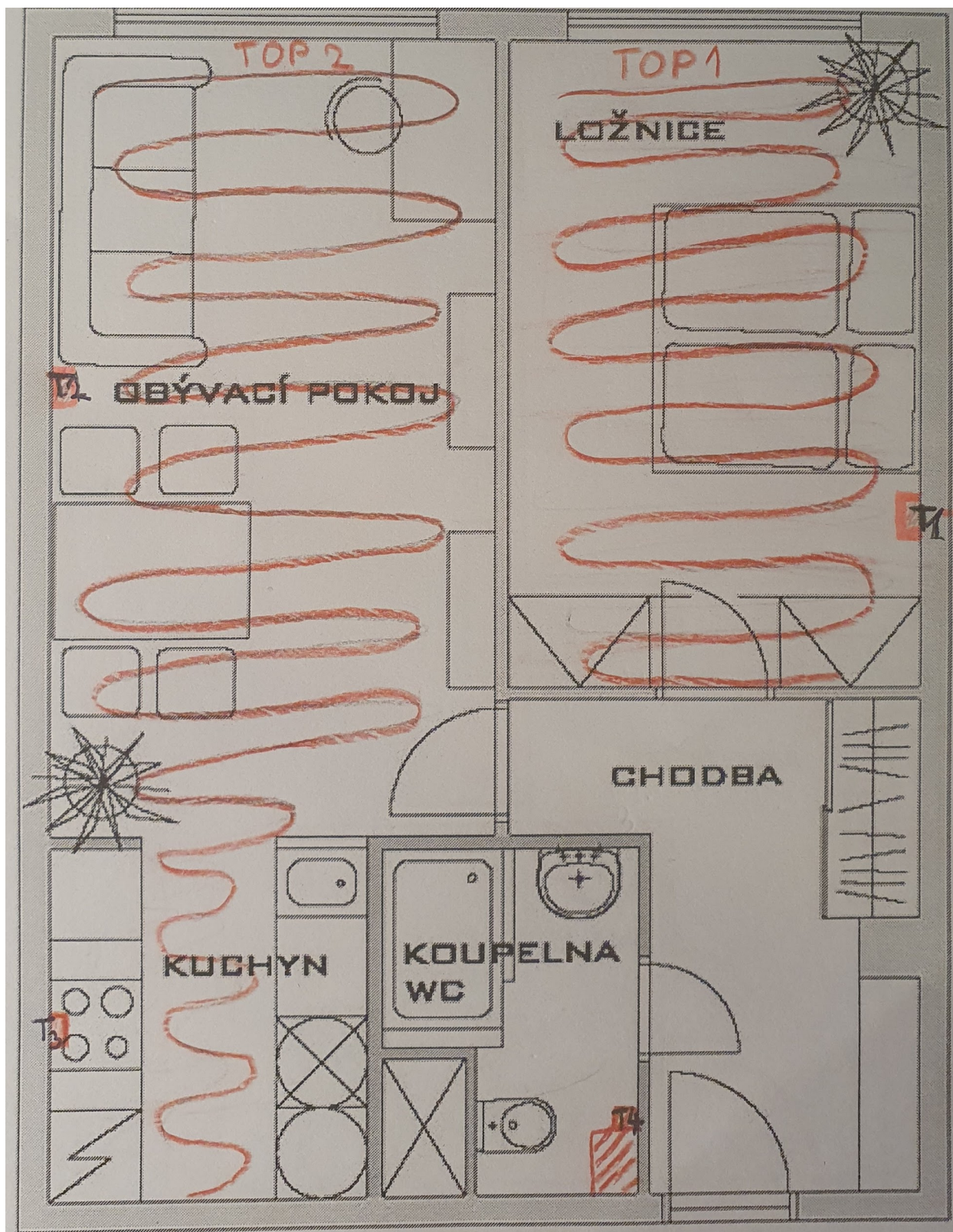
Na vytvoření procesu, co by toto umožňoval nemám momentálně dostatečné znalosti.

S tím jak se práce vyvedla jsem spokojená a jsem ráda za zkušenosti které mi přinesla.

4 Fotografie

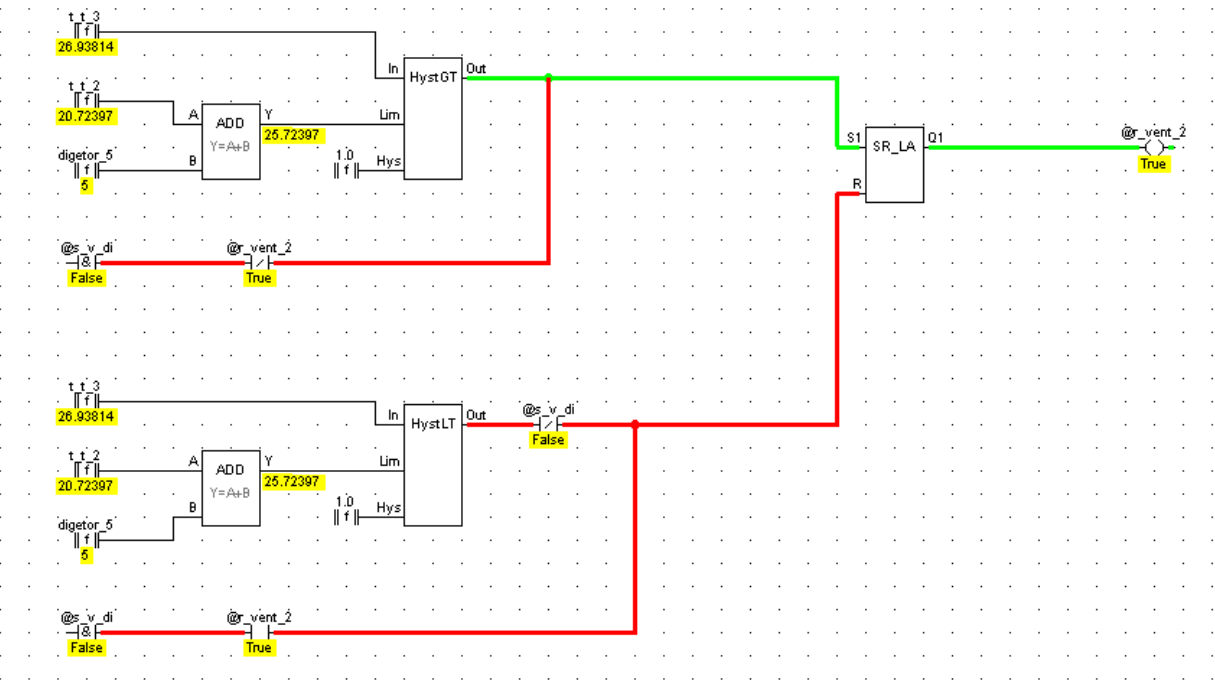


(foto1 - zakreslení rozmístění světel, vypínačů a pohybových čidel)



(foto2 - zakreslení umístění podlahových topení a teplotních čidel)

Ovládání digestoře



(foto4 - ovládání digestoře v režimu *Watch*)

Zdroje :

https://cs.wikipedia.org/wiki/Inteligentn%C3%AD_d%C5%AFm 1.11.2020 17:31

https://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD 1.11.2020 20:13

<https://www.asb-portal.cz/stavebnictvi/technicka-zarizeni-budov/osvetleni-a-elektroinstalace/inteligentni-dum-a-jeho-moznosti> 27.10.2020 18:16

<https://www.insighthome.eu/inteligentni-dum.html> 27.10.2020 18:50

<https://www.nazeleno.cz/bydleni/chytra-domacnost-vyhody-moznosti-a-rizika-inteligentnich-domu.aspx> 28.10.2020 19:15

<https://www.inteligentni-byt.cz/page/jak-to-funguje> 27.10.2020 23:40

<https://www.elektrobock.cz/inteligentni-dum/t2034> 13.11.2020 18:47

<https://www.asb-portal.cz/architektura/rodinne-domy/inteligentni-domy/inteligentni-domacnosti-pomocnik-nebo-hrozba> 13.11.2020 16:23

https://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD 14.11.2020 11:29

<https://www.alza.cz/internet-veci-iot/18858762.htm> 27.10.2020 22:11

<https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie> 14.11.2020 11:43

<https://www.kouba-interier.cz/get.php?id=915> 23.12.2020 8:15

<https://amitomation.cz/en/product/software/detstudio-development-environment/#tab2> 23.12.2020 17:35

http://www.marinfo.cz/Files/ZPT-Vigantice/KATZPT_x/TeplotniCidlaInterierova.pdf 7.1.2021 16:13

https://domat-int.com/wp-content/uploads/2010_07_IDB_Mereni_prostorove_teploty.pdf 7.1.2021 17:15

<https://amitomation.cz/produkt/ridici-systemy/amiris99s/> 8.1.2021 15:26

https://www.zefin.cz/produkty/sensit-snimace-teploty/snimace-teploty/do-interieru-desing-abb/snimace-teploty-tango/ns-101-tango_s982x2018p.html 8.1.2021 17:14

https://www.jabloshop.cz/js-25-combo-detektor-pohybu-osob-a-rozbiti-skla?gclid=CjwKCAiAouD_BRBIEiwALhJH6GZ7xc87L_YjB26fTIxgCrNCDPikH_PB9jQ8cBrtd8d2zPu6KFcQSB0C0YsQAvD_BwE#101 8.1.2021 19:18

https://mojeelektro.cz/kompletni-vypinace-tango-bila/18947-vypinac-tango-abb-jednopolovy-komplet-c1-bily.html?gclid=CjwKCAiAouD_BRBIEiwALhJH6PehFwTdx1HCt4GP99O-GOtGf6BXapqCqXzAePFMqFldNyGKJ1Y_RoC8UkQAvD_BwE 8.1.2021 20:32

[https://mojeelektro.cz/kompletni-vypinace-tango-bila/18951-abb-vypinac-tango-lustrovy-komplet-c5-bily.html?](https://mojeelektro.cz/kompletni-vypinace-tango-bila/18951-abb-vypinac-tango-lustrovy-komplet-c5-bily.html?gclid=CjwKCAiAouD_BRBIEiwALhJH6LHKUSJFowtaJxHbn_GQX_5U6nk0fGDO45u_pqbYt8EOEm9nEhn2hoCdM4QAvD_BwE)

[gclid=CjwKCAiAouD_BRBIEiwALhJH6LHKUSJFowtaJxHbn_GQX_5U6nk0fGDO45u_pqbYt8EOEm9nEhn2hoCdM4QAvD_BwE](https://mojeelektro.cz/kompletni-vypinace-tango-bila/18951-abb-vypinac-tango-lustrovy-komplet-c5-bily.html?gclid=CjwKCAiAouD_BRBIEiwALhJH6LHKUSJFowtaJxHbn_GQX_5U6nk0fGDO45u_pqbYt8EOEm9nEhn2hoCdM4QAvD_BwE) 8.1.2021 21:11

<https://www.elektro-svestka.cz/detail/12203/snimac-teploty-NS-131-SENSIT> 10.1.2021 18:50

[https://www.rele.cz/p/vs116k?](https://www.rele.cz/p/vs116k?gclid=CjwKCAiAi_D_BRAPeiwASslbJ2wYCRPMCizttw9KtTBMeMgY1HyVGaFrqTbaPpIjJD7vCdGa2tV2bhoC5Q8QAvD_BwE#2813)

[gclid=CjwKCAiAi_D_BRAPeiwASslbJ2wYCRPMCizttw9KtTBMeMgY1HyVGaFrqTbaPpIjJD7vCdGa2tV2bhoC5Q8QAvD_BwE#2813](https://www.rele.cz/p/vs116k?gclid=CjwKCAiAi_D_BRAPeiwASslbJ2wYCRPMCizttw9KtTBMeMgY1HyVGaFrqTbaPpIjJD7vCdGa2tV2bhoC5Q8QAvD_BwE#2813) 11.1. 19:25

<http://fatek.seapraha.cz/2008/03/24/zaklladr/> 12.1. 20:37

Program DetStudio jsem stáhla z oficiálních stránek firmy [Amit](#).

Všechny fotografie v dokumentu jsou mé vlastní.