



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

Úsporné bydlení

Economical housing

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vojtěch Marek

VEDOUcí PRÁCE

SUPERVISOR

Mgr. Monika Dosoudilová, Ph.D.

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Vojtěch Marek**
Studijní program: Strojírenství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **Mgr. Monika Dosoudilová, Ph.D.**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Úsporné bydlení

Stručná charakteristika problematiky úkolu:

Návrh systému úsporného bydlení řízeného logickým modulem SIEMENS LOGO!. Úspora bude zaměřena na automatické vypnutí světel, vody, plynu a některých elektrospotřebičů při nepřítomnosti členů domácnosti. Systém bude částečně řešit i komfort bydlení.

Cíle bakalářské práce:

Teorie logického řízení.

Vývojové prostředí LOGO! Soft Comfort.

Návrh systému úsporného bydlení s využitím programu LOGO! Soft Comfort.

Realizace systému s využitím modulu SIEMENS LOGO!

Seznam doporučené literatury:

ŠVARC, Ivan. Automatické řízení. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4398-3.

SIEMENS LOGO! Manual. Industry Support Siemens [on-line]. Munich: Siemens, ©2017 [cit. 2018-0-11]. Dostupné z

https://cache.industry.siemens.com/dl/files/041/109741041/att_924628/v1/logo_system_manual_en-US_en-US.pdf

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Bakalářská práce je zaměřena na vytvoření systému úsporného bydlení za využití kompaktního programovatelného automatu Siemens LOGO!. V práci je zahrnut popis logického řízení a přiblížení problematiky programovatelných automatů. Praktická část se zabývá zjednodušeným návrhem bytové jednotky za využití řídicích programů a samotnou realizaci modelu domečku. K samotné realizaci programu bylo použito prostředí LOGO! Soft Comfort a dále software Autodesk Inventor pro tvorbu součástí potřebných pro model. Práce je doplněna o fotografie samotného modelu a řídicí programy.

ABSTRACT

The bachelor's thesis is focused on the creation of an economical housing system using a compact programmable controller Siemens LOGO!. The work includes a description of logic control and approach to the problem of programmable logic controllers. The practical part deals with a simplified design of a housing unit using control programs and the actual implementation of the house model. The software LOGO! Soft Comfort was used for the actual implementation of the program. Autodesk Inventor software was used for creating of the parts needed for the model. The work is supplemented by photographs of the model itself and control programs.

KLÍČOVÁ SLOVA

úsporné bydlení, automatizace, programovatelný automat, LOGO! Soft Comfort, logické řízení

KEYWORDS

economical housing, automation, programmable controller, LOGO! Soft Comfort, logic control



2021

BIBLIOGRAFICKÁ CITACE

MAREK, Vojtěch. *Úsporné bydlení* [online]. Brno, 2021 [cit. 2021-05-04]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/132931>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Monika Dosoudilová.

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Zdeňkovi Cejpkovi za pomoc a strávený čas při výrobě samotného modelu. Dále bych rád poděkoval paní Mgr. Monice Dosoudilové, Ph.D za vedení bakalářské práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2021

.....

Vojtěch Marek

OBSAH

1	ÚVOD.....	15
2	LOGICKÉ ŘÍZENÍ.....	16
2.1	Logické funkce	16
2.2	Booleova algebra	19
2.2.1	Zákony a pravidla Booleovy algebry.....	19
2.3	Minimalizace logických funkcí	20
2.3.1	Karnaughovy mapy.....	20
2.4	Kombinační obvody	21
2.5	Sekvenční obvody.....	22
2.6	Programovatelné automaty	23
3	PROGRAMOVÁNÍ V PROSTŘEDÍ LOGO! SOFT COMFORT	28
3.1	Vizuální stránka prostředí.....	28
3.2	Konstanty a konektory	29
3.3	Základní funkce	30
3.4	Speciální funkce	31
4	BYTOVÁ JEDNOTKA A NÁVRH ÚSPORNÉHO SYSTÉMU	35
4.1	Osvětlení a automatické rolety	36
4.2	Vytápění místnosti.....	37
4.3	Regulace teploty vody v bojleru.....	38
4.4	Kontrola aktivity v domácnosti	39
5	REALIZACE ÚSPORNÉHO BYDLENÍ.....	40
5.1	Elektrické zapojení	41
5.2	Realizace systému osvětlení a kontrola aktivity v domácnosti	43
5.2.1	PIR senzor.....	44
5.3	Realizace automatizované rolety	45
5.3.1	Světelný senzor.....	46
5.3.2	Řízení stejnosměrného motoru H můstkem.....	46
5.4	Realizace vytápění místnosti	48
5.5	Realizace regulace teploty vody v bojleru.....	50
5.5.1	Optoelektronický snímač	51
6	ZÁVĚR	53
7	SEZNAM POUŽITÉ LITERATURY.....	54
8	SEZNAM PŘÍLOH.....	56

1 ÚVOD

Obsahem této práce bylo využití kompaktního programovatelného automatu Siemens LOGO! pro řízení moderní domácnosti, za účelem snížení nákladů na energie a zvýšení komfortu bydlení. Výsledkem práce byl zmenšený model domečku zahrnující realizaci jednotlivých navrhovaných systémů automatizované domácnosti.

Bakalářská práce je rozdělena do čtyř kapitol. Pro práci s PLC je nezbytná znalost logického řízení. Tomuto tématu je věnována první kapitola. Jsou zde definovány základní pojmy a uvedení do problematiky programovatelných automatů. V druhé kapitole je představeno zvolené programovací prostředí pro tvorbu softwaru. Jedná se o prostředí LOGO! Soft Comfort, které je kompatibilní se zmíněným programovatelným automatem. Třetí kapitola je zaměřena na návrh bytové jednotky, kde by bylo možné realizovat kompletní sestavu řešení. Dále jsou zde uvedeny procentuální úspory, které lze získat zavedením automatizovaných řídicích systémů. Je zde také uveden zjednodušený plán domácnosti. V rámci poslední kapitoly je vyobrazena samotná realizace modelu domečku. Jednalo se o dřevěný model o velikosti 450x400x350 mm, kde byla zahrnuta všechna použitá zařízení. Veškerá zařízení byla řízena vytvořeným programem LOGO! Soft Comfort, který byl následně nahrán do samotného modulu LOGO!. Ke každému systému je uveden samotný řídicí program a použitá zařízení. Některé další potřebné součásti byly vymodelovány v prostředí Autodesk Inventor a následně vytisknuty na 3D tiskárně. Tato kapitola je doplněna o fotografie samotného modelu.

V současnosti je téma automatizovaných domácností velmi oblíbené. Jedná se o častý požadavek při realizaci nového bydlení. Hlavní myšlenka tohoto typu domácností je snížení ročních nákladů za energie. Toto téma jsem si zvolil, protože práce s programovatelnými automaty mě velmi zaujala během studia a výsledné programy lze následně aplikovat i v praxi.

2 LOGICKÉ ŘÍZENÍ

Logické řízení je cílená činnost, při které se logickým obvodem zpracovávají informace o řízeném procesu a následně ovládají příslušná zařízení tak, aby byl splněn předepsaný cíl.

Logickým obvodem chápeme fyzikální systém, který je charakterizován logickými prvky propojenými mezi sebou logickými veličinami. Nejčastější logické veličiny nabývají hodnot logická 0 a logická 1. Hodnotu 0 budeme dále uvažovat v textu jako False (F) nebo vypnuto. Naopak hodnota 1 bude reprezentovat True (T) nebo zapnuto. Mezi těmito stavy nenastává žádný mezistupeň, i když u některých realizací je možné např. otevření napůl. Funkce logického řízení, ale tuto situaci nepředpokládá, a proto tedy nastávají pouze dva možné stavy.

2.1 Logické funkce

V této kapitole se budeme zabývat logickými veličinami nebo logickými proměnnými. Tyto proměnné mohou nabývat pouze konečného počtu hodnot, což považujeme za základní charakteristiku. Na těchto proměnných je založena logická algebra tj. soustava pravidel, určených k popisu vztahů mezi logickými proměnnými. Logickou algebru využíváme nejčastěji při popisu logických operací, které můžeme chápat jako vlastní úkony logické algebry.

Zvláštním druhem logických proměnných jsou dvouhodnotové proměnné, které nabývají pouze dvou možných hodnot. Toto označení je nejčastěji využívané v technice. Jako příklad můžeme uvést napětí není – napětí je, motor neběží – motor běží nebo obecně vypnuto – zapnuto. Logická algebra založená na dvouhodnotových veličinách se také nazývá Booleova algebra. Této problematice se budeme věnovat později.

Logickou funkci

$$y = f(x_1, x_2 \dots x_n) \quad (1)$$

definujeme jako přiřazení hodnot logická 0 (také false nebo nepravda) a logická 1 (true nebo pravda) logické proměnné y ke kombinacím hodnot nezávislých logických proměnných $x_1, x_2 \dots x_n$. V tabulkách místo logická 0 a logická 1 budeme zkráceně psát 0 a 1. Logické funkce mohou být funkce jedné proměnné

$$y = f(x), \quad (2)$$

funkce dvou proměnných

$$y = f(x_1, x_2) \quad (3)$$

a funkce tří a více proměnných, jak vyjadřuje první z výše uvedených rovnic.

Nejjednodušším případem jsou logické funkce s jednou proměnnou. Do této skupiny zařazujeme 4 logické funkce viz. tab.1. V tabulce je uveden název, funkční zápis a pravdivostní tabulka, která pro všechny hodnoty nezávisle proměnné x udává hodnotu závisle proměnné y . Falsum (neboli nulová funkce) je funkce, kde pro libovolné x je

závislá proměnná y rovna 0. Další funkcí je negace. Tato funkce vždy převede hodnotu na hodnotu opačnou. Z možných zápisů budeme používat

$$y = \bar{x} \quad (4)$$

Zápis této funkce čteme, y je negací x nebo y rovná se *non* x . Třetí funkce se nazývá aserce (neboli opakování). Aserce má vždy hodnotu y stejnou jako hodnota x . Poslední funkcí je verum (neboli jednotková funkce). Jak název napovídá, tak pro tuto funkci platí, že hodnota y nabývá vždy stavu 1 pro libovolné x . Pokud se zaměříme na praktický význam, tak nejdůležitější pro nás bude negace, která patří k nejdůležitějším vůbec [1].

<u>Falsum</u> $y = 0$		<u>Negace</u> $y = NOT\ x$ $y = \neg x$ $y = \bar{x}$		<u>Aserce</u> $y = x$		<u>Versum</u> $y = 1$	
x	y	x	y	x	y	x	y
0	0	0	1	0	0	0	1
1	1	1	0	1	1	1	1

Tab. 1: Logické funkce jedné proměnné

Dále se budeme zabývat logickými funkcemi dvou proměnných, které už mají větší praktický význam. Těchto funkcí je celkem 16. Některé z logických funkcí lze opět vyjádřit několika způsoby. Z důvodu snadného zápisu budeme volit ty nejjednodušší způsoby např. u logického součinu místo operátoru AND volíme operátor $*$, který je možný stejně jako v aritmetických výrazech vynechat. Obdobně u logického součtu OR volíme operátor $+$. Některé funkce nemají takový význam pro dvě proměnné, např. opakování, negace nebo falsum. Tyto funkce jsou totožné, jak pro jednu proměnnou, tak pro více proměnných. Dále se tedy zaměříme na ty, které se často používají v řešení problémů.

První z nich je konjunkce (logický součin nebo také AND). Charakteristikou konjunkce je funkční hodnota y , která nabývá hodnotu 1 pouze tehdy, mají-li obě vstupní proměnné x_1, x_2 (obecně všechny vstupní proměnné) hodnotu 1.

Disjunkce (logický součet nebo také OR) nabývá hodnotu logická 1, když alespoň jedna ze vstupních proměnných x_1, x_2 (obecně ze všech proměnných) má hodnotu logická 1.

Negace logického součtu (negace disjunkce, Pierceova funkce, NOR) nabývá logické 1, tehdy když žádná (= ani jedna) z proměnných x_1, x_2 (obecně když žádná z proměnných) nemá hodnotu logická 1.

Jako negací logického součtu, máme také funkci negace logického součinu (negace konjunkce, Shefferova funkce, NAND). Hodnoty logická 1 nabývá tehdy, když proměnné x_1, x_2 (obecně všechny proměnné) nemají současně hodnotu logická 1. Pro zjednodušení, alespoň jedna z proměnných x_1, x_2 musí nabývat hodnoty 0.

Logická funkce ekvivalence nabývá hodnoty 1 v těch případech, kdy vstupní proměnné x_1, x_2 mají stejnou (ekvivalentní) hodnotu tj. obě mají hodnotu 0 nebo 1. Ekvivalenci používáme např. při dokazování tvrzení, že dva logické výrazy mají pro všechny kombinace hodnot svých proměnných stejnou logickou hodnotu.

Dilema (neboli neekvivalence) nabývá hodnotu logická 1 tehdy, když proměnné x_1, x_2 mají různou (neekvivalentní) hodnotu. Tato funkce má význam např. v počítačové grafice.

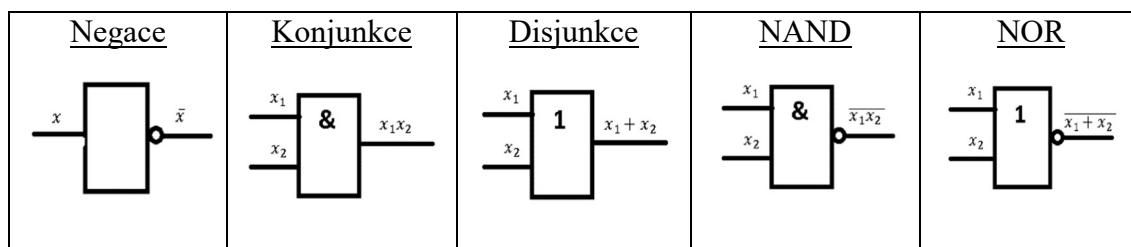
Poslední funkcí, které se budeme věnovat je implikace, nabývá hodnotu logická 1 tehdy, pokud $x_1 \leq x_2$. Předpokládáme přitom, že logická 0 je menší než logická 1. Implikace hraje významnou roli v expertních systémech.

Pokud nás zajímají funkce tří a více proměnných, pouze se opakují funkce dvou proměnných rozšířené na počet, který potřebujeme. Výše zmíněné funkce jsou znázorněny v tab.2, kde podobně jako u funkcí s jednou proměnnou jsou uvedeny názvy funkcí, funkční zápis funkcí a pravdivostní tabulka. Jako příklad, ze skupiny funkcí, kde není rozdíl mezi funkcí jedné proměnné nebo více proměnných, je uvedena negace.

<p><u>Negace</u> $y = \overline{x_2}$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	1	1	1	0	<p><u>Konjunkce</u> $y = x_1 \text{ AND } x_2$ $y = x_1 \wedge x_2$ $y = x_1 * x_2$ $y = x_1 x_2$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	0	1	0	0	1	1	1	<p><u>Disjunkce</u> $y = x_1 \text{ OR } x_2$ $y = x_1 \vee x_2$ $y = x_1 + x_2$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	1	1	0	1	1	1	1	<p><u>NOR</u> $y = x_1 \text{ NOR } x_2$ $y = \overline{(x_1 \vee x_2)}$ $y = \overline{x_1 + x_2}$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	0
x_1	x_2	y																																																													
0	0	1																																																													
0	1	0																																																													
1	0	1																																																													
1	1	0																																																													
x_1	x_2	y																																																													
0	0	0																																																													
0	1	0																																																													
1	0	0																																																													
1	1	1																																																													
x_1	x_2	y																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	1																																																													
x_1	x_2	y																																																													
0	0	1																																																													
0	1	0																																																													
1	0	0																																																													
1	1	0																																																													
<p><u>NAND</u> $y = x_1 \text{ NAND } x_2$ $y = \overline{(x_1 \wedge x_2)}$ $y = \overline{x_1 x_2}$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	1	1	0	1	1	1	0	<p><u>Ekvivalence</u> $y = x_1 \equiv x_2$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	1	<p><u>Dilema</u> $y = x_1 \text{ XOR } x_2$ $y = \overline{(x_1 \equiv x_2)}$ $y = \overline{x_1 \equiv x_2}$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	1	1	0	1	1	1	0	<p><u>Implikace</u> $y = x_2 \rightarrow x_1$ $y = x_1 \leftarrow x_2$</p> <table border="1"> <thead> <tr><th>x_1</th><th>x_2</th><th>y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	1	1	1	1
x_1	x_2	y																																																													
0	0	1																																																													
0	1	1																																																													
1	0	1																																																													
1	1	0																																																													
x_1	x_2	y																																																													
0	0	1																																																													
0	1	0																																																													
1	0	0																																																													
1	1	1																																																													
x_1	x_2	y																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	0																																																													
x_1	x_2	y																																																													
0	0	1																																																													
0	1	0																																																													
1	0	1																																																													
1	1	1																																																													

Tab. 2: Vybrané logické funkce dvou proměnných

Logické funkce používáme pro modelování řízení dvouhodnotových veličin, ale tyto funkce bývají o něco složitější. Toto řízení pak realizujeme pomocí logických obvodů, které zajišťují požadovanou logickou funkci. Pro práci s těmito obvody musíme znát jejich schématické značky. Mezi nejpoužívanější řadíme negaci, konjunkci, disjunkci, negaci disjunktce a negaci konjunktce. Schématické značky jsou znázorněny v tab.3. Pokud je nutné, lze tyto značky modifikovat pro znázornění logických funkcí o třech a více proměnných.



Tab. 3: Schématické značky základních logických funkcí

2.2 Booleova algebra

Booleova algebra nese jméno podle britského matematika George Boolea (1815 - 1864). George Boole žil v první polovině 19. století a zabýval se zjednodušením logiky na jednoduchou logiku dvou stavů. Jednoduchost logiky můžeme chápat jako její vyjádření v co nejmenším počtu logických funkcí. Menší počtem logických funkcí se nám zvyšuje spolehlivost obvodu. Booleova algebra je tedy dvouhodnotová logická algebra, která používá pro popis všech logických funkcí tyto tři základní funkce: negace, logický součet a logický součin [2].

2.2.1 Zákony a pravidla Booleovy algebry

Jak bylo zmíněno výše, pomocí Booleovy algebry minimalizujeme logické funkce. K této minimalizaci je nutná znalost několika pravidel, neboli zákonů. Minimalizaci pomocí Booleovy algebry využijeme při výskytu složitějších logických funkcí. Další metoda minimalizace bude znázorněna v následující kapitole.

Zákony Booleovy algebry:

$$\text{zákon komutativní} \quad x_1 + x_2 = x_2 + x_1 \quad x_1 * x_2 = x_2 * x_1 \quad (5)$$

$$\text{zákon asociativní} \quad x_1 + (x_2 + x_3) = x_1 + x_2 + x_3 \quad (6)$$

$$x_1 * (x_2 * x_3) = x_1 * x_2 * x_3 \quad (7)$$

$$\text{zákon distributivní} \quad x_1 * (x_2 + x_3) = x_1 * x_2 + x_1 * x_3 \quad (8)$$

$$x_1 + x_2 * x_3 = (x_1 + x_2) * (x_1 + x_3) \quad (9)$$

$$\text{zákon absorpční} \quad x_1 + x_1 * x_2 = x_1 \quad x_1 * (x_1 + x_2) = x_1 \quad (10)$$

$$x_1 + \bar{x}_1 * x_2 = x_1 + x_2 \quad x_1 * (\bar{x}_1 + x_2) = x_1 * x_2 \quad (11)$$

$$\text{zákon vyloučeného třetího} \quad x + \bar{x} = 1 \quad (12)$$

$$\text{zákon logického rozporu} \quad x * \bar{x} = 0 \quad (13)$$

$$\text{zákon dvojité negace} \quad \bar{\bar{x}} = x \quad (14)$$

$$\text{zákon opakování} \quad x + x = x \quad x * x = x \quad (15)$$

$$\text{zákon de Morganův} \quad \overline{x_1 + x_2} = \bar{x}_1 * \bar{x}_2 \quad \overline{x_1 * x_2} = \bar{x}_1 + \bar{x}_2 \quad (16)$$

Z výše uvedených zákonů bych rád upozornil na zákon de Morganův. Tento zákon uplatňujeme v algebraických výrazech, kde nám funkce negace přesahuje přes více proměnných. S touto problematikou se setkáváme při převodu Booleovy algebry na logický obvod složený pouze z prvků NAND nebo NOR. Tuto realizaci můžeme nalézt u RS klopných obvodů. Tento typ obvodu nalezneme u mechanických přepínačů, kde zabraňuje nežádoucím zákmitům a následnému rušení signálu. V této oblasti jsou RS klopné obvody velmi spolehlivé.

2.3 Minimalizace logických funkcí

Pro danou logickou funkci můžeme vytvořit několik různých tvarů. Všechny tyto tvary jsou matematicky rovnocenné, protože představují stejnou funkci. Tvarově se mohou tyto funkce značně lišit. Pro každý způsob platí, že každý je z hlediska technického a ekonomického odlišný. Pro technickou realizaci je nutno vždy funkci upravit do nejjednoduššího tvaru, tzn. minimalizovat ji. Cílem minimalizace funkce je dosáhnout co nejmenšího počtu logických prvků (negací, konjunkcí, disjunkcí). Minimalizací se logický obvod stává spolehlivějším, jednodušším a samozřejmě ekonomicky levnějším. V samotném návrhu a realizaci systému úsporného bydlení dále nebylo potřebné logické funkce minimalizovat. Pro tento typ projektu bylo výhodnější průběžné simulování a testování vytvořeného logického obvodu v programovacím prostředí.

2.3.1 Karnaughovy mapy

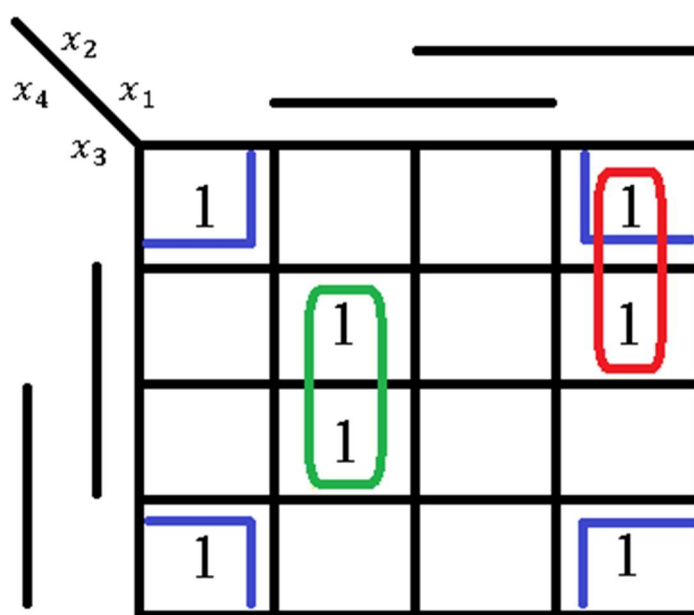
Karnaughovy mapy mohou sloužit jako nástroj pro popis funkce, ale jejich hlavní využití nalezneme v minimalizaci logických funkcí. To je umožněno základní vlastností Karnaughovy mapy a to, že dvě sousední políčka mapy se liší v hodnotě pouze jedné proměnné.

Nejdůležitějším krokem v této metodě je správné sdružování sousedních políček, které obsahují jednotku jako funkční hodnotu. Sdružovat je možné do dvojic, čtveřic, osmic, šestnáctic atd. Smyčky poté přepíšeme do algebraické funkce, kde nezapisujeme ty proměnné, kterým se v rámci smyčky mění hodnota. V některých případech můžeme dále zjednodušit algebraické funkce podle zákonů Booleovy algebry. Ve velké většině případů, ale dostaneme po minimalizaci pomocí Karnaughovy mapy minimální výraz. Často podle zákona vyloučeného třetího ($x + \bar{x} = 1$). Naším cílem je sdružovat sousední políčka do co největších skupin. Sloučením dvou sousedních jednotkových políček vyloučíme jednu proměnnou, sloučením čtyř políček vyloučíme dvě proměnné, sloučením osmi políček tři proměnné atd. Důležité je si ujasnit pojem sousednosti políček v Karnaughově mapě. Zde jsou sousedními políčky např. i políčka na protilehlých

okrajích mapy. Pro zjednodušení této vlastnosti si představme, že tabulku „srolujeme“ jako papír a levý okraj bude sousedit s pravým. Stejný postup můžeme provést i s horním a dolním okrajem, takže dostáváme další možnosti pro sdružování políček.

Základní pravidla pro minimalizaci logických funkcí Karnaughovými mapami :

- Všechny jedničky v mapě musí být zakroužkovány, žádnou nesmíme vynechat.
- Každá jednička se může při kroužkování použít několikrát (to umožňuje zákon opakování $x + x + x + \dots = x$).
- Přednost mají větší smyčky před menšími. Nejmenší prioritu mají osamocené jedničky.
- V rámci pravidla, podle kterého žádnou jedničku nesmíme vynechat, se snažíme o co nejmenší počet smyček [1].



$$y = \bar{x}_1 \bar{x}_3 + x_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_4$$

Obr. 2.1 Příklad Karnaughovy mapy

2.4 Kombinační obvody

Jedním ze dvou typů logických obvodů jsou kombinační logické obvody. Jejich hlavním znakem je jednoznačnost funkčních hodnot na výstupu, které jsou určeny kombinací hodnot vstupních proměnných. Z toho nám vyplývá, že neobsahují žádnou operační paměť, takže nezávisí na předchozích stavech obvodu [3]. Obecně tedy platí, že právě jedné vstupní kombinaci vstupních proměnných odpovídá právě jedna výstupní kombinace. Signál se tedy šíří pouze jedním směrem od vstupu do výstupu. Matematicky bychom mohli kombinační obvod popsat :

$$\begin{aligned}
 y_1 &= f(x_1, x_2 \dots x_n) \\
 y_2 &= f(x_1, x_2 \dots x_n) \\
 &\dots \\
 y_n &= f(x_1, x_2 \dots x_n)
 \end{aligned}
 \tag{17}$$

Kombinační obvody se zpravidla skládají z logických hradel. U logických hradel, ale musíme počítat se zpožděním, které závisí na technologii výroby hradel, struktuře obvodu a především na délce tzv. kritické cesty, což je nejdelší posloupnost od vstupu k výstupu. Typickými příklady kombinačních obvodů jsou dekodéry, multiplexory, komparátory a obvody pro aritmetické operace.

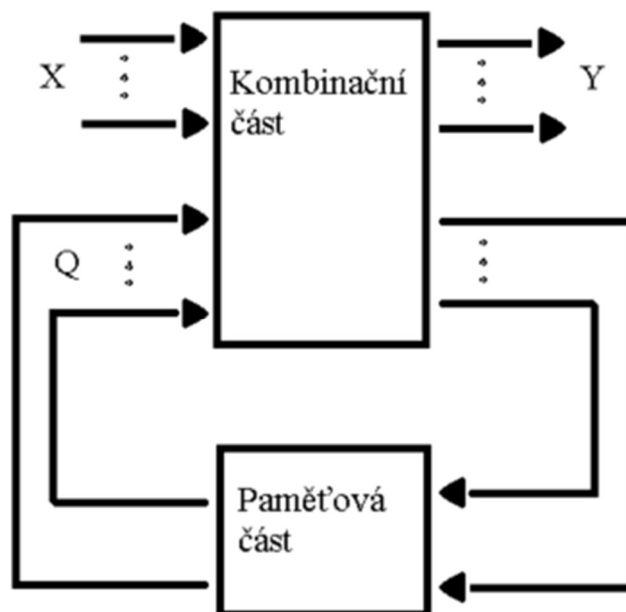


Obr. 2.2 Blokové schéma kombinačního obvodu.

2.5 Sekvenční obvody

Druhou variantou, jak můžeme realizovat logický obvod, je sekvenční obvod. Hlavní rozdíl mezi kombinačním a sekvenčním obvodem spočívá v určování výstupní proměnné, obecně výstupu. U sekvenčních obvodů záleží nejen na kombinaci vstupních hodnot, ale také na jejich časově předcházejících kombinacích hodnot [3]. Sekvenční obvody obsahují paměťovou část, kde jsou předcházející kombinace ukládány do následujícího okamžiku.

Sekvenční obvody dělíme na synchronní a asynchronní. U synchronních je každá změna vstupních a výstupních proměnných řízena synchronizačními impulsy, které zajišťují stejné okamžiky změn všech proměnných. U asynchronních tomu tak není a změny jsou vázány pouze na změny vstupních proměnných. Na obr. 2.3 je znázorněno blokové schéma obecného sekvenčního obvodu, kde množina X označuje vstupní proměnné, množina Y výstupní proměnné a množina Q označuje vnitřní (stavové) proměnné. Závislost na předchozím stavu je zajišťována tak, že hodnoty (některých) výstupních proměnných jsou vedeny zpět na vstup.



Obr. 2.3 Blokové schéma sekvenčního obvodu

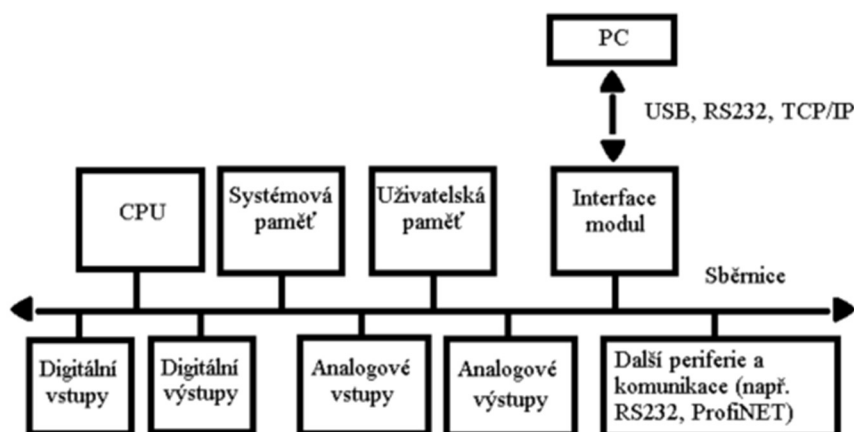
Sekvenční obvody mohou realizovat jak jednoduché operace, tak i složitější zapojení jako např. mikroprocesory. Své využití naleznou také v jiným aplikacích jako např. klopné obvody, registry, čítače nebo paměti.

2.6 Programovatelné automaty

Programovatelné automaty jsou programovatelné řídicí systémy umožňující řízení průmyslových a technologických systémů a procesů v reálném čase [1]. Každý automat je vybaven různě vyspělým hardwarem i softwarem, což nám dovoluje využívat různé automaty pro speciální situace. Základní modely běží na 8/16 bitových jednočipových mikroprocesorech (MCU), výkonnější modely se naopak už dají srovnávat s dnešními CPU v počítačích třídy PC. Programovatelné automaty můžeme označovat zkratkou PLC (Programmable Logic Controller), která je používána častěji než celý název. Běžný programovatelný automat obvykle umožňuje rozsáhlé možnosti komunikace (vstupy/výstupy, GSM, TCP/IP protokol, externí paměti, HMI rozhraní apod.), regulace (integrované PID/PSD regulátory, fuzzy řízení, automatické nastavování regulátorů apod.). Se současnou výbavou jsou PLC schopna téměř libovolné realizace disktrétních číslicových systémů. Pro lepší představu o těchto systémech uvedme jako obecný příklad regulátory. Programovatelné automaty dělíme na dva základní typy. Prvním typem jsou kompaktní PLC a druhým modulární. Častěji se potkáme s modulárně řešenými automaty, což znamená, že jednotlivé komponenty můžeme měnit, přidávat, přestavovat atd. Jako příklad modulárně řešeného automatu uvedeme skupinu automatů od firmy Siemens SIMATIC. Jako příklad kompaktního automatu si uvedeme logický modul Siemens LOGO!.

Výhodou programovatelných automatů je jejich odolnost v různých typech prostředí. Jejich práci neovlivní vlhké nebo prašné prostředí, což dovoluje jejich využití v různých odvětvích. V současné době představují PLC nejrozšířenější typ řídicího systému v průmyslové praxi, v oboru techniky budov i v mnoha dalších aplikačních oblastech [5]. Obecně je řešení problému využitím PLC finančně výhodné. Cena hardwaru je přijatelná v poměru cena/výkon. Poslední výhodou je nenáročnost na obsluhu, která souvisí se spolehlivostí programovatelných automatů.

Vnitřní strukturu PLC bychom mohli přirovnat k architektuře počítače. Základem je systémová sběrnice, která spojuje jednotlivé bloky modulárního PLC. Tyto bloky propojujeme podle potřeby pomocí jedné nebo více sběrnic. Pokud bychom se zaměřili na kompaktní PLC, tak spojování více modulů nemusíme řešit. Podle schématu modulárního PLC (viz. obr. 2.4) si ukážeme základní jednotky PLC. Jako běžné modulární jednotky považujeme centrální procesorovou jednotku, která potřebuje systémovou a uživatelskou paměť, dále nalezneme interface pro propojení s PC a moduly pro digitální/analogové vstupy a výstupy. Napájení automatu je obvykle na hodnotě 24V stejnosměrného napětí. Pokud bychom napájeli střídavým napětím, tak obvyklou hodnotou je 230V.



Obr. 2.4 Schéma modulárního PLC

Hlavní komponentou celého programovatelného automatu je centrální procesorová jednotka (CPU, Central Processor Unit). Procesorová jednotka určuje výkon programovatelného automatu. Můžeme se setkat s jednoprocessorovými, ale i s víceprocesorovými jednotkami. V rozdělení, která CPU jsou výkonnější, hraje velkou roli operační rychlost odvozená z tzv. doby cyklu. Doba cyklu je dána jako čas, za který zpracuje CPU 1000 logických instrukcí [4]. Dnešní CPU se pohybují v rozmezí desítek ms až desetin ms. Dalším parametrem, který hraje roli v centrální jednotce je paměťový prostor. Z výše uvedeného schématu vidíme dvě různé paměti. Uživatelská paměť ukládá uživatelský program a systémová systémový program. Systémový program obsahuje veškeré uživatelské registry, čítače, časovače a systémové proměnné. Obě paměti bývají typu EPROM a dokáží uložit řádově desítky KB a jednotky MB. Zkratkou EPROM

označujeme statickou a energeticky nezávislou paměť, která má mazatelný obsah, což je její největší devízou. V samostatné jednotce lze přidat navíc uživatelskou paměť. Všechny CPU jsou vybaveny základními logickými operacemi a souborem instrukcí pro aritmetiku.

PLC má dvě varianty možných vstupů, digitální vstupy/výstupy (logické, binární, dvouhodnotové) a analogové vstupy/výstupy. K digitálním vstupům se běžně připojují tlačítka, přepínače, koncové spínače a další snímače s dvouhodnotovým charakterem signálu. Digitální vstupy zajišťují ochranu všech vstupů PLC před poškozením nebo zničením, odfiltrování krátkodobých rušivých impulsů, galvanické oddělení obvodů vstupního modulu od CPU a signalizaci stavu vstupů [4]. Digitální výstupy plní podobné funkce jako vstupy, pouze s rozdílem, že zesilují signál na potřebnou úroveň. Analogové vstupy a výstupy naopak zajišťují kontakt PLC se spojitými veličinami. Nejdůležitější součástí vstupů je A/D převodník, který převádí analogové signály na číslicové. Typickým příkladem analogových jednotek jsou teplotní snímače (odporové, polovodičové nebo termočlánky), snímače tlaku nebo snímače vlhkosti. Využití analogových výstupů najdeme například u servoměniče, který lze tímto způsobem ovládat.

Komunikační jednotky, jak název napovídá, umožňují komunikaci mezi moduly vstupů a výstupů se systémy, popřípadě podsystémy, počítači atd. Komunikace je velmi důležitá u modulárních PLC, kde jsou moduly od sebe vzdáleny. Existují i jednotky, které poskytují komunikaci přes modem nebo telefonní síť. Komunikační jednotky využívají různé komunikační rozhraní např. RS 232, RS 422 nebo RS 485.

Programovatelné automaty mohou pracovat v různých časových chodech programů. Typické pro PLC je tvz. cyklické spouštění programu. Princip tohoto spouštění je v neustálém cyklickém provádění úkonu dokola. Pokud bychom se zaměřily na rozdíly mezi PLC a osobním počítačem, tak právě vykonávání programu v cyklu je typické pro programovatelné automaty. Cyklické spouštění, ale není jediný možný způsob, jak lze pracovat s PLC. Další možností je tvz. časové spouštění programu. Program je spouštěn v definovaný časový okamžik. Tento způsob je nutný pro regulační algoritmy. Poslední možností je tvz. alarmové spouštění programu. Program je spouštěn, pouze při inicializaci nežádoucí změny na vstupní jednotce. Tento způsob je určen pro rychlé reakce a upozornění obsluhy stroje. Alarmově spouštěný program má nejvyšší prioritu z všech zmíněných. Dokáže přerušit, jak časové, tak cyklické spouštění. Naopak nejnižší prioritu má cyklické spouštění.

Programování PLC realizujeme pomocí existujících specializovaných jazyků, které byly původně navrženy pro realizaci logických funkcí. Programovací jazyky jsou standardizovány podle mezinárodní normy IEC EN 61191-3, která specifikuje pět typů jazyků pro programování PLC. Jazyky rozdělujeme do dvou základních skupin, textové jazyky a grafické jazyky. Normu nelze brát jako zajištění kompatibility mezi všemi programovatelnými automaty od všech výrobců.

Podle mezinárodní normy IEC EN 61191-3 dělíme jazyky následovně: Jazyky reléových (kontaktních) schémat (Ladder Diagram, LD). Jde o grafický programovací jazyk. Program se zobrazuje ve formě schémat používaných při práci s reléovými a kontaktními prvky. Výhodou je snadné hledání chyb pomocí vodivých cest a srozumitelnost jazyku pro širší spektrum oborů [1]. Ladder Diagram je vhodný pro zpracování velkého počtu logických signálů. Naopak problémy mohou nastat u složitějších úloh, kde se může stát nepřehledným. Méně vhodný je také pro aritmetické operace a práci s ASCII znaky [6].

Dalším ze skupiny grafických jazyků je jazyk funkčních bloků, též logických schémat (Function Block Diagram, FBD). Základní logické operace popisuje obdélníkovými značkami. Funkční bloky zprostředkovávají velmi dobrou názornost a přehlednost programu. Další výhodou je srozumitelnost pro ostatní odvětví průmyslu, kde lze jednoduše v těchto programech číst. Znovu může nastat problém u složitějších projektů, tak jak je typické pro grafické jazyky. Příkladem FBD je LOGO! Soft Comfort.

Posledním ze skupiny grafických jazyků je jazyk sekvenčního programování (Sequential function Chart, SFC). Program má formu specializovaných bloků sekvenčního chování řídicího programu. Je odvozen ze symboliky Petriho sítí [1]. K popisu se používají značky stavů, přechodů a větvení. Kroky a přechody musí mít asociovány logické podmínky. Bez těchto podmínek by nemohl program proběhnout. Tento zápis je velmi efektivní s dokumentací a své využití nalezne ve vytváření hlavní větve programu, kde se s SFC setkáme nejčastěji. Nevhodný je pro přímou realizaci zpracování analogových signálů [6].

Jazyk instrukcí (Instruction List, IL) je obdobou assembleru u počítačů. Je strojově orientován, to znamená, že každé instrukci odpovídá stejně pojmenovaný příkaz jazyka [1]. Spadá do skupiny textových jazyků. Tyto jazyky jsou často používané, zejména profesionálními programátory. Textové jazyky jsou obecně vhodnější pro velikostně větší projekty, ale s tím je spojena také větší nepřehlednost a orientace v programu. Velká výhoda spočívá v paměťové i rychlostní úspornosti.

Jazyk strukturovaného textu (Structure Text, ST) je obdobou vyšších programovacích jazyků. Svě základy čerpá z jazyků Pascal a C. Stejně jako IL, patří mezi textové jazyky. Umožňuje úsporný a názorný zápis algoritmů. Z praktického hlediska je dnes nejrozšířenější alternativou moderních programovatelných automatů [1]. Je vhodný pro programování složitých aritmetických operací a zpracování analogového signálu. Jako nevýhodu lze považovat nutnou znalost syntaxe a potřebných příkazů, ale tato nevýhoda je velmi individuální. Jako příklad můžeme uvést programovací prostředí Automation Studio od firmy B&R automation, které pracuje s ST.

Dále se zaměříme na realizaci programu, ukázkou využitého programovacího prostředí a návrh úsporných systémů. Před praktickou částí bylo nutné se seznámit s pojmem programovatelný automat, abychom si dokázali představit, jak pracuje a proč je vhodné v našem případě zvolit právě programovatelný automat. Práce probíhala na logickém modulu od firmy Siemens LOGO!, který má výhodu v dostupnosti a příznivé cenové relaci. Dalším důvodem proč byl zvolen tento modul je možná realizace

v moderních domácnostech. Typické pro tento modul je zapojení do elektrických rozvodů. K programování jsem si zvolil jazyk funkčních bloků (FBD), který považuji za nejpřehlednější a nejvhodnější pro řešení automatizace moderních domácností. Hlavním pozitivem jazyka je schopnost naprogramovat potřebné úkony i v domácích podmínkách a následně aplikovat bez odborného dohledu.

3 PROGRAMOVÁNÍ V PROSTŘEDÍ LOGO! SOFT COMFORT

V této kapitole se blíže seznámíme s prostředím LOGO! Soft Comfort [7,8].

Prostředí LOGO! Soft Comfort, od firmy Siemens, pracuje s jazykem funkčních bloků FBD. Podle mezinárodní normy programovacích jazyků IEC EN 61191-3 spadá FBD do skupiny grafických jazyků. LOGO! Soft Comfort je součástí zakoupených balíčků s modulem LOGO!, ale je možné vyzkoušet aplikaci v demoverzi. Prostředí je určeno pro vytváření programů obvodů v co nejnázornější formě. Výsledný program je pouze schéma zapojení ve formě funkčních bloků, které jsou propojeny pomocí drag and drop. Drag and drop neboli „táhni a pusť“ označuje druh operace či ovládání v rámci grafického uživatelského rozhraní v různých programech a aplikacích [9]. Obecně jde o klasické grafické prostředí, které je přívětivé k uživateli. Díky neustálým aktualizacím softwaru je vylepšováno, podle potřeb uživatelů nebo postupujícímu trendu.

Prostředí je schopné provést simulaci, bez toho abychom museli mít propojený modul s programem. Výhodou je možnost simulace jak digitálních, tak analogových signálů, např. teploty, intenzity osvětlení, vodního průtoku, vlhkosti atd. Veškeré tyto signály lze realizovat a následně s nimi v rámci modelu pracovat. V nastavení simulace je možné použít časové spouštění programu i cyklické spouštění programu. Pojmy časové spouštění programu a cyklické spouštění programu byly definovány v podkapitole programovatelné automaty.

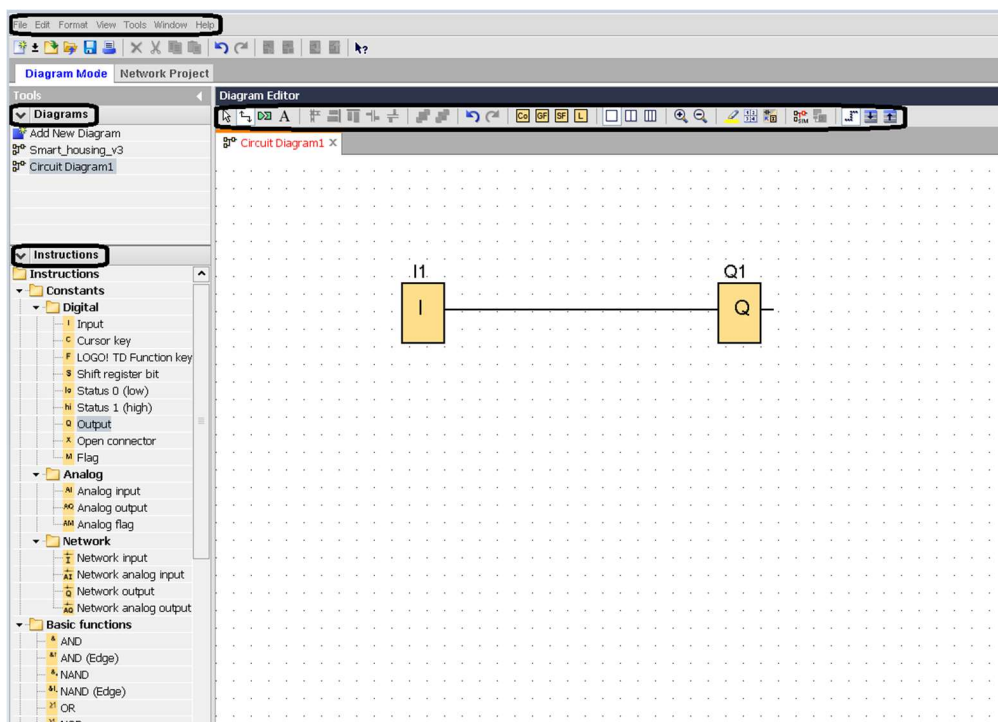
3.1 Vizualní stránka prostředí

Vizuální stránka není nejdůležitější částí programovacího prostředí, ale je potřeba si ukázat základní úkony s aplikací. V levé horní části nalezneme klasický přehled možností pro práci s programem jako file, edit, format, view, tools, window a help. Veškeré změny programu jako jeho uložení, přepsání nebo úpravu formátu lze nalézt v těchto záložkách.

Na levé straně obrazovky máme nástroje diagrams a instructions. V oblasti Diagrams vidíme otevřené projekty a lze mezi nimi libovolně přeskakovat, popřípadě přidat další, které potřebujeme mít v rychlém přístupu. V Instructions vyhledáváme funkční bloky, následně vybereme a vložíme do pracovní plochy. Nejčastěji používané funkční bloky budou ukázány dále.

Nejdůležitější část obrazovky zaplňuje pracovní plocha, neboli diagram, kde realizujeme samotný program. Pro příjemnější práci a rychlejší přístup jsou v horní liště diagramu vloženy ikony k často používaným funkcím: šipka pro pohybování s funkčními bloky, nástroj pro vytváření spojů, uspořádání a přiblížení okna, změna velikosti plochy, spuštění simulace nebo online ladění programu [10].

Celkový vzhled prostředí je prezentován na obr. 3.1. Jako ukázka bylo zvoleno spojení vstupu a výstupu pro lepší představu, jak vypadá propojení cest v prostředí.



Obr. 3.1 Ukázka prostředí LOGO! Soft Comfort

3.2 Konstanty a konektory

Konstanty a konektory jsou vstupy, výstupy, příznaky a pevné hodnoty napětí (konstanty). Všechny programy potřebují nějaký vstup a následně nějaký výstup. Stejně jako na fyzickém modulu PLC má prostředí digitální a analogové vstupy/výstupy. Programovací prostředí LOGO! Soft Comfort pracuje se vstupy i výstupy jako s konstantami, ale je možné vkládat funkční bloky reprezentující přímo hodnotu požadované konstanty.

Digitální vstupy jsou označovány pomocí písmene I viz obr. 3.1. Číslo, které doplňuje digitální vstup, reprezentuje vstupní konektor fyzického modulu LOGO!. Z principu funkce digitálních konektorů vyvodíme, že pokud přivedeme na příslušný vstup napětí, je příslušný blok aktivní. Analogové vstupy jsou označeny písmeny AI a analogicky s digitálními vstupy k nim přísluší číslice, které znovu reprezentují fyzický konektor k modulu. Pro náš dostupný fyzický modul je možné pracovat pouze se dvěma analogovými vstupy, ale na trhu jsou dostupné i moduly s více analogovými vstupy.

Výstupy jsou také dvojího typu a odpovídají výše zmíněným vstupům. Digitální výstupy označujeme písmenem Q a znovu připadá k písmenu číslice, která koresponduje s číslicí konektoru modulu LOGO!. Pomocí písmen AQ označujeme analogové výstupy, které opět můžeme využít maximálně dvakrát, z důvodu dvou možných fyzických výstupů na modulu. Modul LOGO! má k dispozici ještě 16 prázdných výstupů, které nelze používat opakovaně, ale své využití najdou např. pro speciální funkce „Textové zprávy“. Tyto výstupy jsou označovány písmenem X.

Speciální skupinou jsou příznakové bloky, které jsou označeny písmeny M nebo AM. Označení M platí pro digitální příznaky a AM pro analogové příznaky. Jde o virtuální výstupy, které mají hodnotu odpovídající svému vstupu. Tyto příznakové bloky často využíváme pro porovnávání hodnot, mazání atd.. K dispozici máme 24 digitálních příznaků a 6 analogových příznaků. Příznak M8 je tzv. inicializační příznak, který je po provedení prvního cyklu programu resetován a následně ho lze využít pro nastavování, mazání a vyhodnocování jako ostatní příznaky.

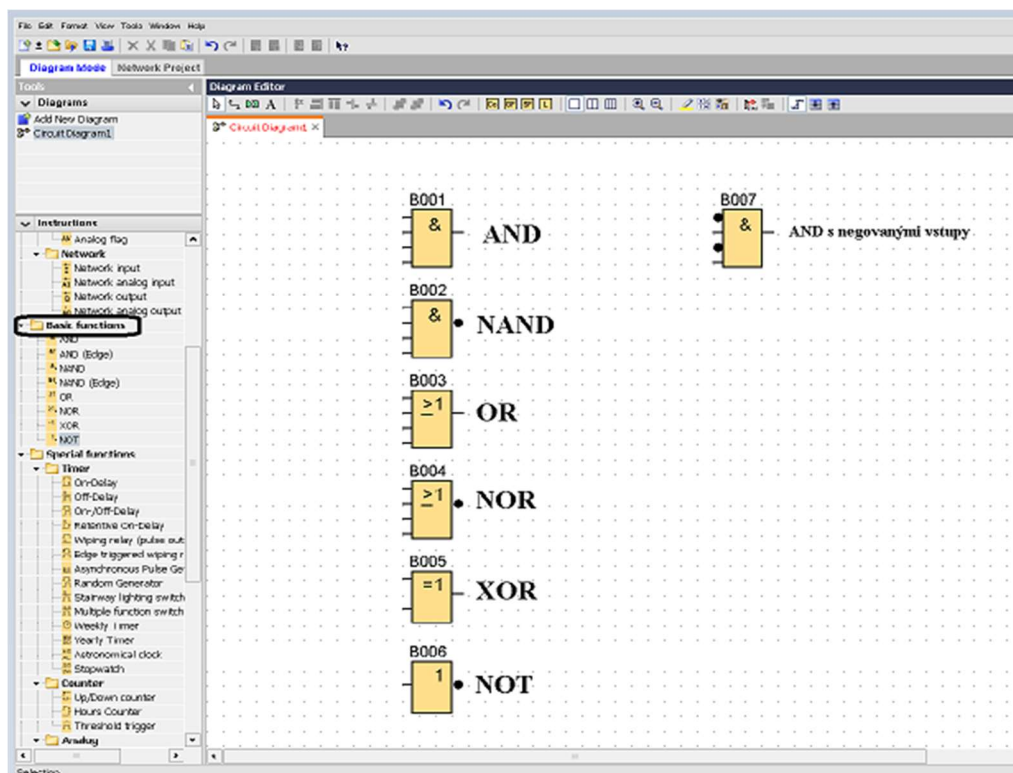
3.3 Základní funkce

Základní funkce v prostředí LOGO! Soft Comfort odpovídají základním logickým funkčním prvkům jako AND, OR a NOT. V tomto prostředí jsou tyto funkce znázorněny velmi podobně, jako schématické značky (viz. Tab.3). Vstupy jednotlivých základních funkcí můžeme invertovat, tj. program invertuje logickou 1 na příslušném vstupu na logickou 0 [7]. Dále jsou uvedeny základní funkce a jejich aplikace v programovacím prostředí.

Funkce AND pracuje maximálně se čtyřmi vstupy. Výstup funkce je pouze jeden. Podmínkou pro aktivní výstup funkce AND je hodnota logické 1 na všech čtyřech vstupech, ale je možné pracovat i s menším počtem vstupů, díky nastavení bloku v prostředí. Pokud na některý ze vstupů nepřivádíme signál, blok pracuje s prázdnými vstupy jako hodnotou 1. Mezi funkčními bloky nalezneme také tzv. AND s vyhodnocením hrany. Tento funkční blok je vhodný pro aplikace s náběžnou nebo sestupnou hranou signálu. S funkčním blokem AND souvisí další základní funkce NAND. Chování bloku NAND je očekávané a jde tedy o negaci logického součinu. V prostředí LOGO! Soft Comfort obě funkce vypadají vzhledově velmi podobně, pouze se liší plným kolečkem na výstupu prvku, které značí negaci. Negovat lze také vstupy bloků pomocí dvojitého kliknutí levým tlačítkem na příslušný vstup nebo použitím prvku NOT, který provede negaci hodnoty, ale v některých případech může být nadbytečný.

Další důležitou funkcí logických obvodů je logický součet OR, který má také své zastoupení ve zvoleném programovacím prostředí. Chování bloku je typické pro logický součet. Stav výstupu prvku OR je 1 pouze v případě, je-li alespoň jeden vstup 1, tj. nejméně jeden z kontaktů je sepnut [7]. Znovu lze pracovat maximálně se čtyřmi vstupy. S prvkem OR dále souvisí funkční bloky NOR a XOR. Funkční blok NOR reprezentuje negaci logického součtu a znovu ho nalezneme v nabídce bloků. Prvek XOR souvisí s nonekvivalencí vstupů. Výstup prvku XOR nabývá 1, tehdy když vstupy nejsou ekvivalentní.

Z ukázky základních funkčních bloků, lze vidět že zvolené prostředí je velmi intuitivní a jednotlivé názvy prvků se shodují s očekávaným chováním dle definic logických prvků v kap. Logické prvky. Symboly funkčních bloků v prostředí se mohou trochu lišit od schématických značek logických prvků, proto je uvedena grafická ukázka základních bloků na obr. 3.2. V instrukcích nalezneme tyto funkce v záložce Basic functions.



Obr. 3.2 Základní funkce a jejich symboly.

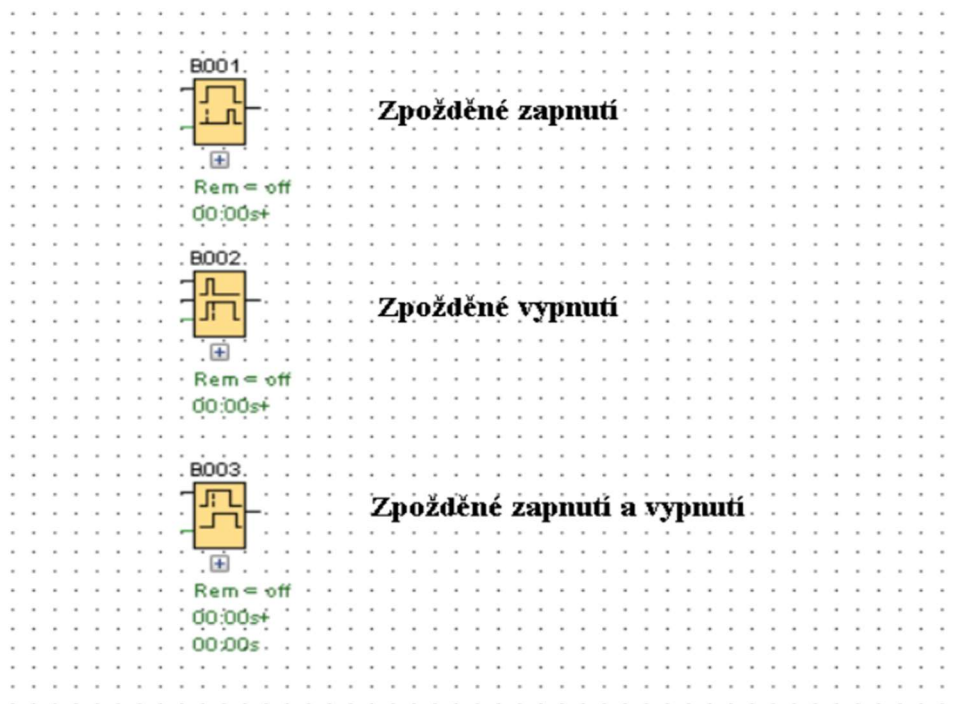
3.4 Speciální funkce

V programovacím prostředí LOGO! Soft Comfort je velké množství různých typů funkcí. Do skupiny speciálních funkcí patří různé spínače, automatické hodiny, čítače nebo přímo PI regulátor. Dále budou zmíněny pouze funkce, které jsou nejčastěji využívány v následné realizaci programu. Rozdíl mezi základními funkcemi a speciálními je v označování vstupů. Obecně jsou tyto vstupy značeny příslušnými znaky nebo zkratkami, které označují funkci daného vstupu. Obvykle lze ve speciálních funkcích nastavovat parametry pro lepší přizpůsobení programu. Velkou výhodou speciálních funkcí je vnitřní nastavení bloku. U základních funkcí jsou tato nastavení neuskutečnitelná.

Všechny tyto funkce je nutné u klasických PLC ručně naprogramovat. V prostředí LOGO! Soft Comfort máme k dispozici 40 funkčních bloků, které nahrazují jednotlivé úkony. Pokud neexistuje požadovaný prvek je možné si vytvořit vlastní funkční blok a vytvořit si specifickou knihovnu těchto bloků.

Speciální funkce zpožděné zapnutí sepne výstup se zpožděním, pokud vyprší specifikovaný čas. Doba, o jakou má být výstup zpožděn, je nastavena v parametru T a je konfigurovatelná. Pokud během odečítání času proběhne změna na vstupu opět na 0, bude čas znovu resetován. Funkčně opačný prvek zpožděné vypnutí, po uplynutí specifikovaného času resetuje výstup. Čas lze znovu nastavit pomocí parametru T. Pokud je nutné, lze využít funkční blok zpožděné zapnutí/vypnutí. Funkce tohoto prvku spočívá v zapnutí výstupu po uplynutí nastavené doby a následném resetování po uplynutí doby pro zpožděné vypnutí. Jde o spojení dvou výše zmíněných prvků, a proto jsou nutné dva

nastavitelné parametry T_H a T_L . První parametr určuje dobu zpožděného zapnutí a druhý parametr specifikuje čas pro zpožděné vypnutí. Všechny zmíněné funkční bloky jsou nejčastěji využívány např. u osvětlení reagující na pohyb nebo automatického otevírání a zavírání dveří.



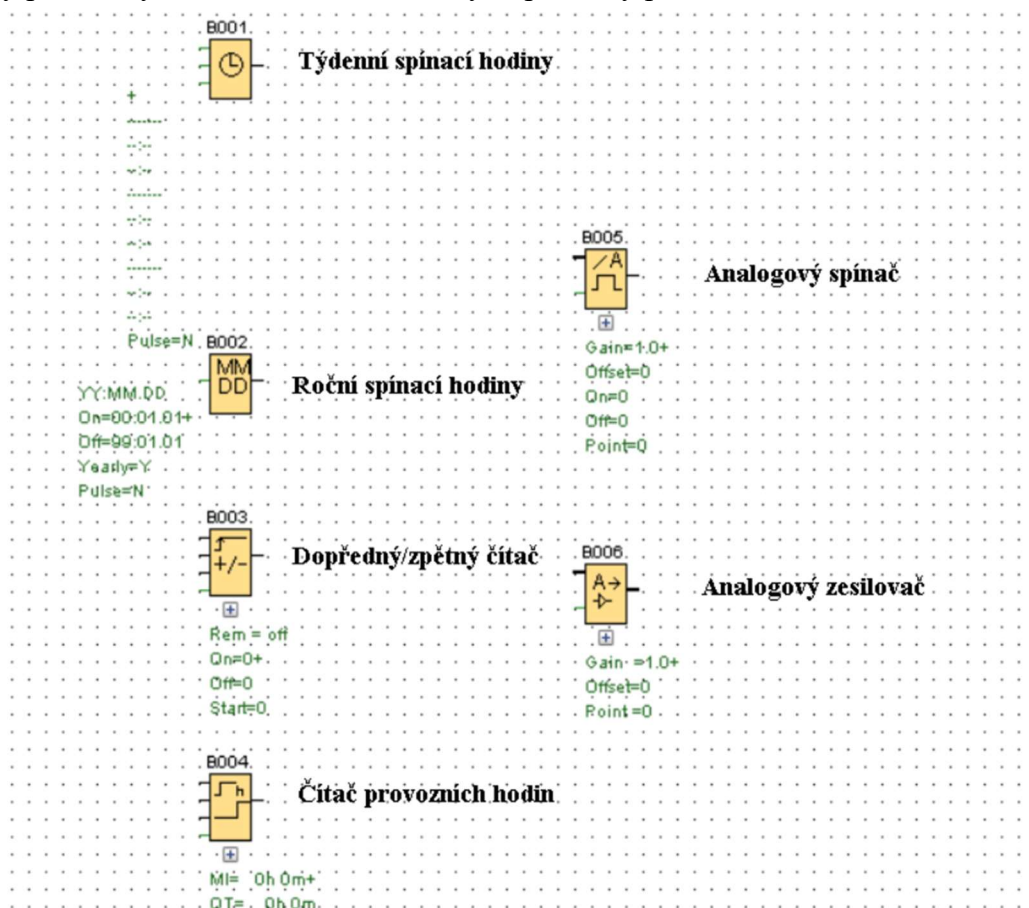
Obr. 3.3 Funkční bloky ovlivňující dobu výstupu

Týdenní a roční spínací hodiny řídí výstup zadaným datem zapnutí a vypnutí. Obě funkce podporují jakoukoliv kombinaci dnů resp. měsíců. U týdenního spínání lze nastavit až 3 spínací časy např. každý den od 5:00 do 10:30 h, každé pondělí od 14:15 do 15:00 h a každou sobotu od 16:00 do 16:35 h. Roční spínače jsou nastaveny podle dne zapnutí a vypnutí. Nevýhodou je původní nastavení bloku, které způsobí resetování prvku po zadaném dnu vypnutí. Z tohoto důvodu je nutné zavést dva a více ročních spínačů, pokud chceme vícekrát do roka určité zapnutí. Výstupy těchto bloků poté můžeme spojit pomocí funkčního bloku OR.

Důležitými speciálními funkcemi jsou čítače. Prostředí LOGO! Soft Comfort obsahuje dva druhy funkčních bloků pro čítač. První variantou je dopředný/zpětný čítač. Podle nastavení parametru je při každém pulzu na vstupu zvětšena nebo zmenšena vnitřní hodnota. S čítači lze pracovat i na výstupu, kde po dosažení nastavené hraniční hodnoty lze výstup sepnout nebo restartovat. Druhá možná varianta nese název čítač provozních hodin. Při signálu na monitorovaném vstupu En začne ubíhat zadaný čas a po uběhnutí tohoto času bude sepnut výstup.

Pro práci s analogovými signály využíváme další speciální funkční bloky jako např. analogový spínač, analogový zesilovač atd. Výstup analogového spínače je zapínán a resetován podle dvou nastavitelných prahových hodnot [7]. V tomto prostředí nalezneme i další typy analogových spínačů např. analogový rozdílový spínač nebo

analogový komparátor. Oba zmíněné funkční bloky pracují také s nastavitelnými prahovými hodnotami, ale také s diferenční hodnotou popřípadě s rozdílem dvou vstupujících analogových hodnot. Veškeré tyto spínače najdou uplatnění v aplikacích pro hlídání požadovaných hodnot teploty, vlhkosti, intenzity osvětlení apod. Analogový zesilovač ukládá procesní proměnnou analogového vstupu do paměti, zapíná výstup a přičte požadované posunutí podle zvoleného módu [7]. Analogový zesilovač lze nastavit do různých módů jako např. PT100/PT1000 pro práci s teplotami. Po nastavení jsou změněny parametry funkčního bloku tak, aby odpovídaly požadavkům.



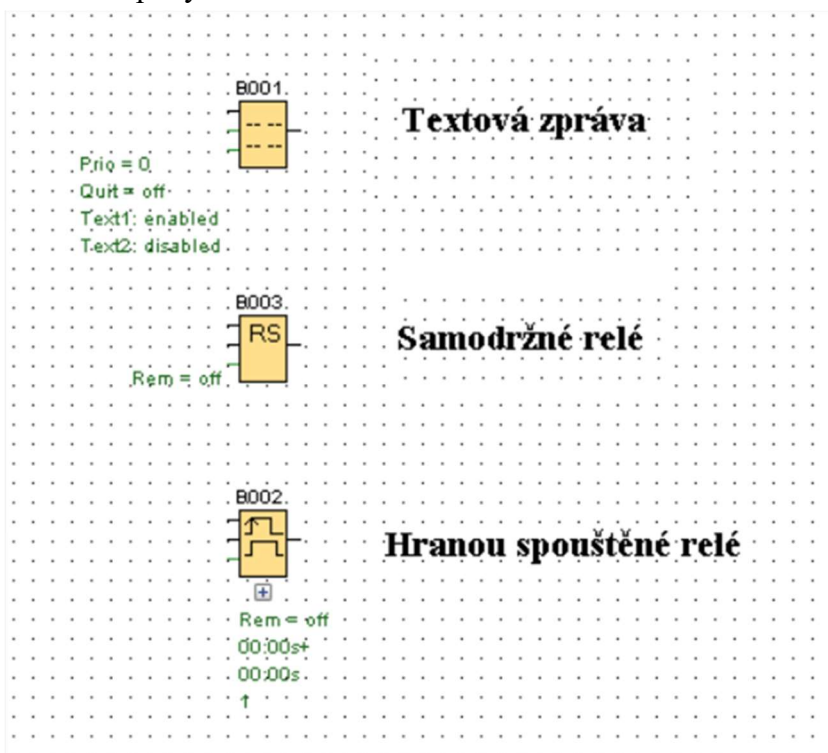
Obr. 3.4 Funkční bloky spínacích hodin, čítačů a bloků pro práci s analogovým signálem

Funkce textové zprávy, zobrazena na displeji modulu LOGO!, slouží ke komunikaci s uživatelem. V rámci nastavení funkčního bloku, zde můžeme napsat textové zprávy, které chceme aby uživatel viděl, nebo k textu navázat jiný funkční blok, kde lze pozorovat změny hodnot. Textová zpráva je zobrazena na displeji modulu, pokud je na vstup prvku přivedena logická 1. V celém programu lze využít maximálně 10 funkcí textové zprávy. Mezi jednotlivými zprávami lze poté na modulu LOGO! přeskokovat pomocí kláves nahoru a dolů.

Pokud požadujeme neměnný výstup po delší dobu, využijeme funkční blok samodržného relé. Tato speciální funkce reprezentuje RS klopný obvod. Klopný obvod tohoto typu má dva vstupy: nastavovací S (set) a resetovací R (reset). Impuls přivedený

na vstup S klopného obvodu způsobí jeho překlopení ze stavu 0 do stavu 1. Jestli je klopný obvod již ve stavu 1, pak se nestane nic. Účinek impulsu na vstup R je opačný [11]. Na totožném principu funguje funkční blok v programovacím prostředí. V nastavení bloku lze pouze měnit remanenci. Zjednodušeně to lze shrnout do volby mezi možnostmi bez/s remanence/cí. Nastavení remanence hraje roli v případě, pokud modul není napájen nebo nastane výpadek sítě a my vyžadujeme uložení aktuálního výstupního stavu.

Funkční blok hranou spouštěné relé reaguje na náběžnou hranu vstupního signálu. Po zaznamenání vstupního signálu generuje blok výstupní pulz, který má předvolenou šířku TH a mezeru mezi pulzy TL. Parametry lze znovu nakonfigurovat v nastavení bloku. Své využití najde blok u zařízení, která mají „mrtvý“ okamžik kdy nereagují např. při opakovaném snímání pohybu.



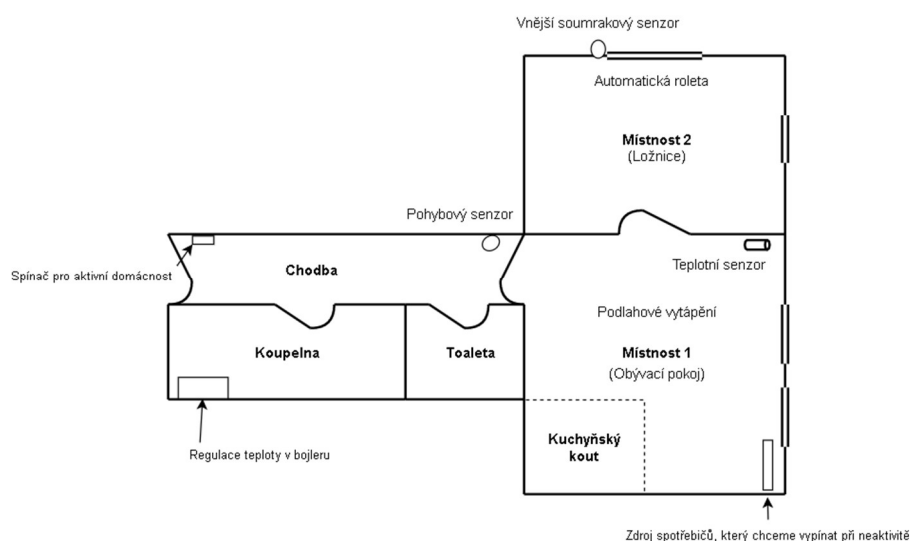
Obr. 3.5 Funkční bloky textové zprávy, samodržného relé a hranou spouštěného relé

4 BYTOVÁ JEDNOTKA A NÁVRH ÚSPORNÉHO SYSTÉMU

Za pomoci kompaktního modulu Siemens LOGO! je možné realizovat chytrou domácnost se základními požadavky na automatizovanou domácnost. Mezi tyto požadavky lze zahrnout regulaci teploty v místnosti, vody nebo automatizované řízení osvětlení v domě. Pro účely této práce je tento modul dostačující. Kapitola se bude zabývat návrhem bytové jednotky, kde by bylo možné aplikovat řídicí systém a možné benefity oproti klasické domácnosti. Systém je zaměřen na úsporu energie, která úzce souvisí s finanční stránkou. Všechny uvedené návrhy jednotlivých automatizovaných systémů lze zavést do současných domácností.

Pro lepší představu uvažované domácnosti je uveden zjednodušený plán bytové jednotky. Bytová jednotka obsahuje dvě hlavní místnosti, kuchyňský kout, koupelnu, toaletu a vstupní chodbu. Přesné rozměry bytové jednotky jsou v tomto případě zanedbatelné. Ve skutečnosti je ale velmi důležité pracovat i s rozměry jednotlivých pokojů. Typickým příkladem může být teplota v místnosti. Podle velikosti místnosti je nutné nastavit dostatečně vysokou teplotu pro rychlé vytápění apod. Zjednodušený plán domácnosti je zobrazen na obr. 4.1.

Pro ukázkou poplatků za energie jsou brány průměrné hodnoty pro českou tříčlennou domácnost v roce 2020. V rámci poplatků za elektřinu se jedná o cenu 4,76 Kč za 1kWh a roční spotřebu 2200 kWh ročně [12]. Náklady za vytápění v návrhu zapadají také do elektrické spotřeby, protože dále bude pracováno s elektrickým podlahovým vytápěním. Proto je nutné přičíst průměrně 9500 Kč za vytápění. Jedná se o přibližnou částku zjištěnou z různých zdrojů pro byt o velikosti 2+kk (kuchyňský kout). Výsledné poplatky za elektřinu jsou 19972 Kč ročně.



Obr. 4.1 Plán navržené bytové jednotky

4.1 Osvětlení a automatické rolety

Součástí každé domácnosti je systém osvětlení místností. Řešení těchto systémů je velmi různorodé. Setkat se můžeme se stropním osvětlením, stolními lampami nebo stojacími podlahovými lampami. Stojací podlahové lampy jsou často pořizovány za účelem dekorace místnosti. Nevýhodou těchto lamp je větší energetická spotřeba a malý výběr z variant, které lze automatizovat nebo konfigurovat jejich spotřebu elektrické energie. V návrhu bylo řešeno pouze stropní osvětlení, které dostatečně zlepší viditelnost v místnosti a je možné zajistit úsporu. Samozřejmě jde dále domácnost doplnit o stolní lampy u pracovních stolů, které návrh nezahrnuje.

Přesné číslo o kolik je možné ušetřit na poplatcích za elektřinu díky správnému nastavení osvětlení nelze přesně uvést. Podle různých zdrojů se lze pohybovat v rozmezí mezi 15 až 20 procenty. Veškeré tyto hodnoty jsou velmi subjektivní a odlišné pro každou domácnost podle hodin strávených ve své domácnosti. Ale z těchto čísel lze vyzorovat vliv osvětlení na poplatky za elektrickou energii. Roli v samotných úsporách hraje také volba typu žárovky. Dále k vyšším úsporám může napomoci využití potenciálu stavby jako např. využití slunečního svitu v oblasti.

Návrh systému pracuje se vstupním osvětlením v chodbě, kde je vhodné využít pohybový senzor. Důvodem je zdržování osob pouze po krátkou dobu. Pro samotnou realizaci je stěžejní práce s pohybovým senzorem. Tento typ senzoru snímá pohyb v určitých vzdálenostech. Při zaznamenání pohybu sepne svůj výstup a následně se může rozsvítit osvětlení. Pro obývací pokoj je doporučováno využít radarový senzor, který reaguje citlivěji než senzor pohybu. Radarový senzor je schopný regulovat intenzitu osvětlení v souvislosti s denním osvětlením a četností pohybu v místnosti [13]. Senzor pohybu je obvykle součástí radarového senzoru. Časový interval, po jakou dobu zůstane žárovka rozsvícena, lze měnit podle požadavků v samotném řídicím programu. Otázka spočívá ve volbě typu žárovky. Kvůli častému zapínání a vypínání osvětlení je důležité zvolit žárovku, která vydrží velký počet spínacích cyklů. V tomto případě byla vybrána LED žárovka, která vydrží vysoký počet spínacích cyklů a teoreticky má delší životnost.

S osvětlením musíme brát v potaz intenzitu venkovního světla. V mnoha situacích není nutné využívat elektrické osvětlení. Nejčastěji se s touto situací lze setkat v letních dnech, kdy slunce zapadá až v pozdních večerních hodinách. V tomto návrhu je pro tento účel použit světelný senzor v rámci systému automatizovaných rolet. Jedná se o reakci na venkovního intenzitu světla a následný pohyb rolet. Rolety jsou aktivní pouze v požadovanou dobu. Nastavení pracovní doby je individuální, ale není vhodný pohyb rolet v nočních hodinách. Při nečekané reakci senzoru může dojít k pohybu rolet a to může vést k nežádanému probuzení uživatele. Automatizované rolety nejen napomáhají k nižším poplatkům za osvětlení, díky využití denního světla, ale řeší také komfort bydlení. Některé systémy reagují i na dálkové ovládání, ale v tomto případě se jedná o plně automatizovaný návrh. Oba systémy jsou podmíněné aktivitou v domácnosti.

4.2 Vytápění místnosti

Vytápění bytových jednotek je dalším faktorem při rozhodování jestli zavést automatizovanou domácnost. Variant jak vytopit dům, byt či chatu je mnoho. Nejčastěji se setkáváme s topením přírodními materiály, elektrickým vytápěním či klasickými článkovými radiátory. Ve všech případech teplotu v místnostech ovlivňuje více faktorů. Základem dobré udržitelnosti teploty je kvalitní izolace bytu či domu. Pro příklad lze uvést možné finanční ztráty ve starších domech. Ve špatně zaizolovaném starším rodinném domě 4+1 stojí vytápění i přes 40 tisíc korun ročně [15]. V návrhu je brán v potaz dobře zaizolovaný byt v novostavbě. Dále je uvažováno nové zateplení stavby a nová okna, která nedovolují samovolný únik tepla. Dalšími vlivy na vytápění mohou být pozice bytové jednotky vzhledem k slunečnímu záření nebo rozměry místností. Úspora na vytápění, kterou lze zajistit zavedením automatizovaných systémů vytápění se pohybuje kolem 20 procent.

Navržený systém je zaměřen na podlahové vytápění místnosti. V posledních letech se podlahové vytápění stalo fenoménem v moderních domácnostech. V současných novostavbách narazíme často na realizaci právě tohoto typu vytápění. Důvodem, proč se stalo tak oblíbené, je pocit tepla přímo od nohou. Obyvatelé nabývají většího pocitu tepla v místnosti. V takovém případě není nutné topit tak často a v takové míře. Nevýhodou může být delší doba pro dosažení žádané teploty po spuštění vytápění [14]. Podlahové vytápění řeší znovu také komfort bydlení.

V návrhu vytápění místnosti je tepelným zdrojem topná fólie. Tato fólie bývá součástí dnešních podlahových vrstev. Dále je nutné zavést teplotní senzor pro sledování teploty např. PT 1000. Výhodou elektrických topných fólií je velmi účinná regulace teploty v jednotlivých místnostech. V laboratorních podmínkách, při samotné realizaci, si ale vystačíme s rezistory jako zdrojem tepla. Cílem systému je automatická regulace teploty v místnosti za přítomnosti obyvatel a následném přechodu do úsporného režimu, při jejich delší absenci. Samotné řízení probíhá v modulu LOGO!. Systém podle sledované teploty spouští nebo vypíná topnou fólii. Udržovaná teplota je nastavena na 23°C, což je příjemná pokojová teplota ve všech ročních obdobích. Ideální místností pro automatizovanou regulaci teploty je obývací pokoj. Pokud systém nezaznamenává aktivitu v domě, podlahové vytápění přejde do úsporného režimu, který udržuje teplotu na nižší hodnotě 21 °C. Úsporný režim je zařazen z důvodu úspor elektrické energie, kdy neustálé zapínání/vypínání vytápění může mít kontraproduktivní účinek. Také lze uvažovat přítomnost domácích mazlíčků, kterým tato teplota může vyhovovat. Velikost teploty lze nastavit podle potřeby uživatele pro oba režimy a není nutné se řídit výše uvedenými hodnotami. Automatický přechod do úsporného režimu je řízen systémem kontroly aktivity v bytové jednotce, který bude vysvětlen dále.

Pokud využíváme podlahové vytápění je potřeba velmi opatrně pracovat s teplotou topné fólie. Z principu funkce vytápění z podlahy je jasné, že teplo pocítujeme zprvu od nohou. Proto teplota nášlapné vrstvy nesmí přesáhnout 29 °C. Překročení této

hodnoty by mohlo mít negativní vliv na lidský organismus [15]. Toto doporučení je třeba dodržovat při tvorbě řídicí aplikace či realizace podlahového vytápění v domě.

S teplotou v místnosti dále souvisí také větrání. Je samozřejmostí, že není možné udržet stejnou teplotu a neotevřít si přitom okno. Návrh systému pracuje s automatizovaným větráním. Jako větrání lze využít automatické otevírání oken nebo zavést samotnou vzduchotechniku v bytě. Řídicí program má nastavené přesné časy, kdy dojde k větrání. Jde o kratší ale intenzivní okamžiky, aby došlo k provzdušnění místnosti a nedocházelo k problému s vydýchaným vzduchem. Pokud je větráno, není efektivní topit. Proto je systém nastaven na zamezení vytápění v období, kdy je aktivní větrání. Hodiny i časové intervaly větrání jsou znovu nastavitelné v samotném řídicím programu podle potřeb uživatele.

4.3 Regulace teploty vody v bojleru

Ohřev vody je velmi dobře kontrolovatelný proces. Pokud jsme schopni zajistit kontrolu teploty vody a dobu, kdy požadujeme ohřev, tak jsme schopni ušetřit obvykle 20 procent z poplatků za teplou vodu. Tento fakt však nemusí platit pro všechny domácnosti. Velikost úspor se odvíjí od typu a kvality pořízených řídicích zařízení pro kontrolu teploty vody. Dále může hrát roli samotný typ stavby. V nízkoenergetických či pasivních domech většinou potřebujeme více energie na ohřev vody a tato úspora by proto nebyla tak efektivní [15]. Návrh systému opět pracuje s předpokladem klasického bytu, popřípadě menšího domku s chytrým bojlerem, který reguluje teplotu vody.

Chytré bojler se v oblasti elektrického ohřevu vody začínají objevovat v posledních pár letech. Nejde o nic jiného než běžný elektrický bojler, který je doplněn o pokročilou řídicí a komunikační jednotku. Tyto bojler lze nastavovat podle potřeb uživatele, jako např. v pracovní dny v určitou hodinu ohřívat vodu [17]. Většinou tato zařízení komunikují přes mobilní aplikaci apod. To se netýká našeho návrhu systému pro programovatelné automaty. Tato chytrá zařízení mohou díky inteligentní elektronice sledovat spotřebu teplé vody a následně informovat uživatele na mobilní telefon nebo PC. Z dostupných testů si lze udělat srovnání mezi běžným a chytrým bojlerem. Chytrý bojler oproti běžnému obstál v testech lépe a zajistil roční úsporu 1400 až 1700 Kč [18].

Návrh systému je zaměřen na regulaci teploty vody v bojleru. Cílem aplikace je automatizovaná regulace teploty na požadované hodnotě. Laboratorní simulace bude probíhat za pomoci zařízení pro vzduchovou levitaci. V trubici bude kontrolována poloha plastového míčku. Výška míčku bude reprezentovat hodnotu teploty. Následně proběhne reakce větráku, který bude simulovat ohřev vody. Pokud klesne míček pod zadanou hodnotu, větrák se automaticky spustí. Jakmile míček znovu dosáhne požadované výšky, větrák se automaticky vypne. Princip udržování míčku okolo jedné hladiny je analogický s principem udržováním teploty vody. Tato realizace v rámci laboratorní úlohy postačí. Způsob simulace byl zvolen dle možností univerzitní laboratoře. Poloha míčku bude snímána optoelektronickým senzorem, který sleduje plastový míček. Systém je nastaven, aby po prvotním spuštění, či restartování automaticky spustil větrák („ohřál vodu“),

dokud se míček nedostane na požadovanou polohu. Není tedy nutné manuálně spouštět ohřev. Funkce ohřevu je znovu podmíněna přítomností osob v domácnosti. Kompletně automatizované řízení zajistí větší míru úspor. Pokud se v domácnosti nikdo delší dobu nenachází, není důvod dále regulovat teplotu vody.

4.4 Kontrola aktivity v domácnosti

Veškerá funkčnost chytré domácnosti se odvíjí od aktivity jejích členů. Pokud obyvatelé nejsou přítomni po delší časový interval nebo odjíždí na týdenní dovolenou, není důvod udržovat domácnost v plném fungování. Obecně lze zajistit tímto krokem úspory na více frontách. Pokud jsme schopni takový systém zavést, pokles poplatků se projeví na spotřebě elektrické energie, teplé vody, topení i plynu. Přesná čísla reprezentující úsporu tohoto systému nelze uvést, protože jde o velmi subjektivní systém a záleží na volbě připojených zařízení.

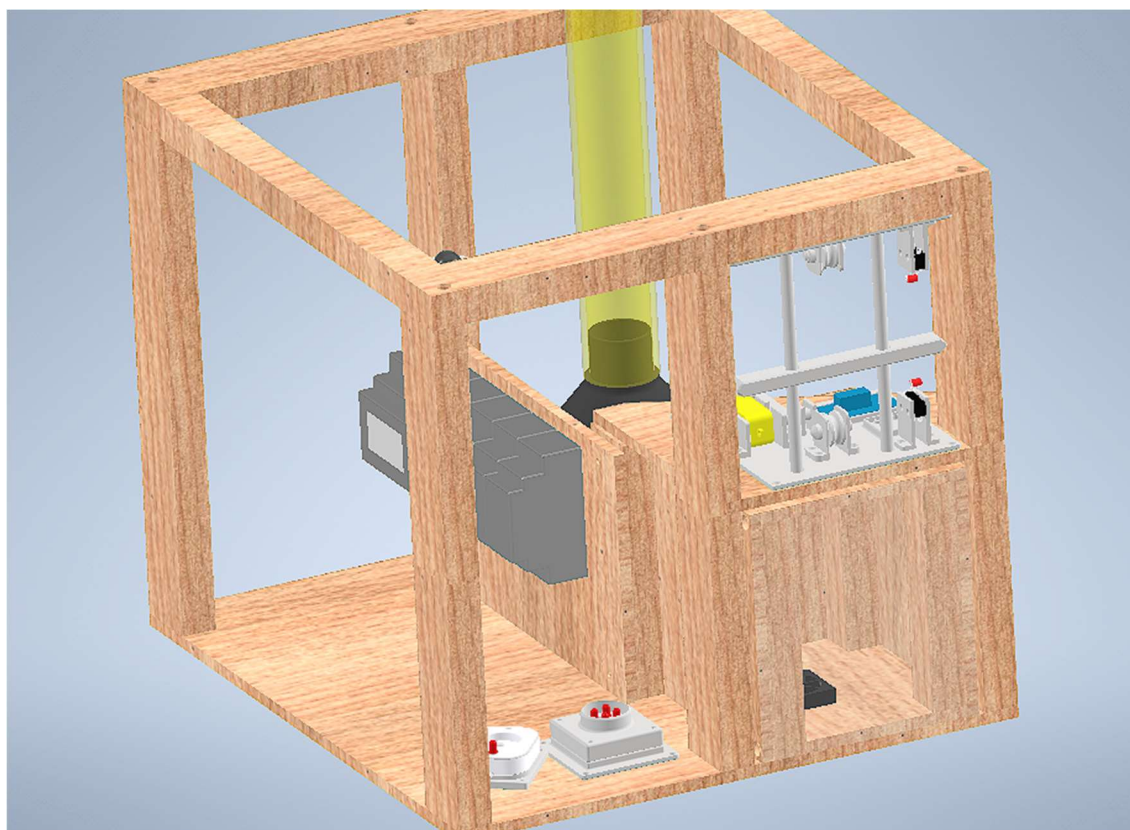
Návrh systému můžeme považovat jako takovou podmínku pro plnou funkčnost všech ostatních systémů v domácnosti. V tomto případě se jedná o všechny výše zmíněné systémy chytré domácnosti. Osvětlení v chodbě, automatické rolety či regulace teploty vody v bojleru jsou aktivními pouze pokud se nějaká osoba zde nachází. Systém vytápění je také navázán na kontrolu aktivity, ale formou přechodu mezi jednotlivými režimy vytápění. Navrhovaný systém je zaměřen hlavně na úsporu elektrické energie. Veškerá zařízení v domácnostech potřebují většinou elektrickou energii, a proto je třeba cílit na tuto oblast. Systém lze cílit také na jiné oblasti jako např. plyn. V rámci neaktivity by šlo o utažení kohoutků pro přívod plynu.

Kontrola probíhá v prostoru chodby. Řídící program pracuje s četností osvětlení a následné reakci čítače provozních hodin. Pokud je zaznamenáno rozsvícení osvětlení dojde k resetování zmíněného čítače. Chodba je ideálním prostorem, kde lze zavést tento typ kontroly. Podle plánu bytové jednotky vidíme, že pouze z chodby se lze dostat jak na toaletu, tak do koupelny. Pokud nějaká osoba domácnost obývá, je chodba klíčovým prostorem. Tato skutečnost je dostatečným vodítkem pro volbu vhodného prostoru. Výstupem rozhodování je pouze možnost je/není někdo doma. Časový interval pro neaktivitu v domě je nastaven na 20 hodin. Pokud během 20 hodin není zaznamenáno rozsvícení světel v chodbě, domácnost se automaticky změní do pasivního režimu. Přejít domácnosti do pasivního režimu znamená změnu v chování jednotlivých systémů. Hlavní reakcí na přechod do pasivního režimu je automatické vypnutí zásuvek spotřebičů. Ideálními spotřebiči, které lze tímto způsobem vypínat jsou televizory, kuchyňské spotřebiče (krom lednice), zábavní elektronika nebo pokojové lampy.

Při návratu do domácnosti nepřechází vše automaticky do aktivního režimu. Pasivní domácnost změníme zpět spínačem, který je nutný stisknout při návratu. Spínač je umístěn na stěně za vstupními dveřmi. Díky automatizovanému osvětlení se zde nachází pouze jeden spínač, aby se předešlo problémům v přehlednosti spínačů. Ruční spouštění systémů z pasivního režimu předchází možným chybám v automatickém samovolném spuštění.

5 REALIZACE ÚSPORNÉHO BYDLENÍ

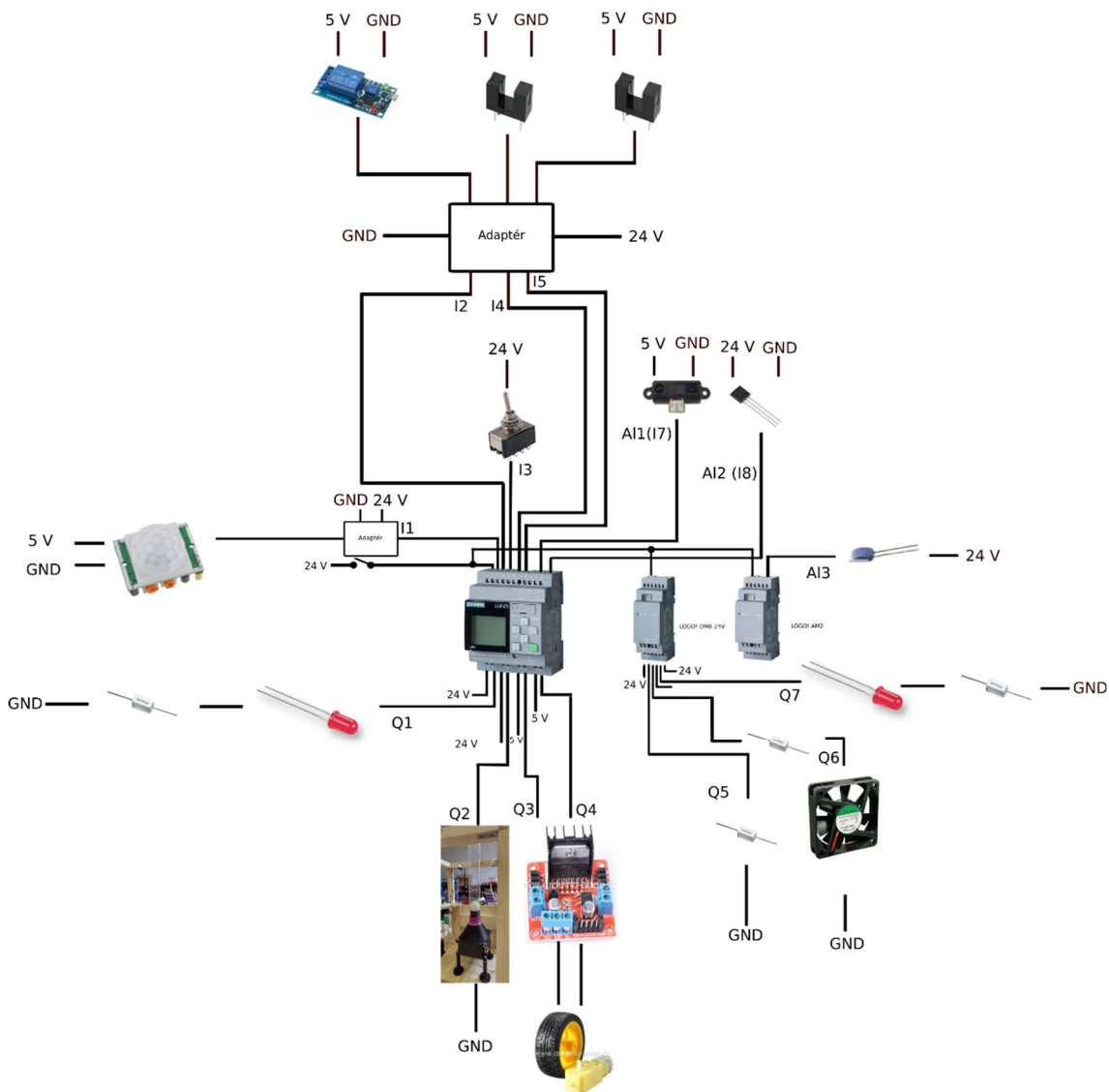
Kapitola je zaměřena na samotnou realizaci navrhovaných systémů a přesný popis použitých zařízení. Řídící aplikace byla naprogramována v prostředí LOGO! Soft Comfort, které je zaměřeno na jazyk funkčních bloků. K řízení systému byl použit kompaktní modul LOGO!. K modulu byly připojeny potřebné rozšiřující moduly. Bytová jednotka bude reprezentována zmenšeným dřevěným modelem domečku. Použitá zařízení jsou modelového typu a znázorňují funkčnost jednotlivých programů. Taková zařízení jsou k vidění např. v RC modelech. Ve skutečnosti jsou k dispozici větší a výkonnější zařízení, která jsou používána v chytrých domácnostech. K přípravě modelu domečku a tvorbě některých potřebných držáků byl použit software Autodesk Inventor. V tomto prostředí byl samotný model navržen a průběžně kontrolován viz obr. 5.1.



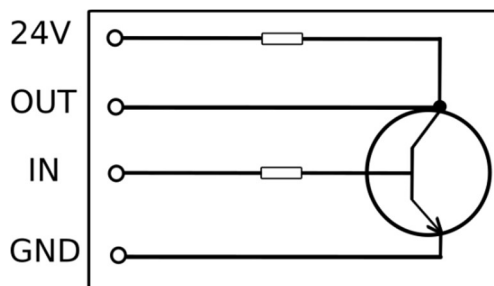
Obr. 5.1 Model domečku v prostředí Autodesk Inventor

5.1 Elektrické zapojení

Zjednodušené elektrické schéma zapojení je na obr. 5.2. Je zde znázorněno potřebné napájení jednotlivých zařízení a dále potřebná zapojení pro rozšiřující modul. Pro některá zařízení bylo nutné zajistit napájení odlišné než typických 24V. Jednalo se o zařízení napájena 5V. Pro tato zařízení byl pořízen měnič napětí z 24V na 5V. Dále tato zařízení pracují většinou s 3,3V logikou, proto bylo nutné kvůli vstupům modulu LOGO! vyrobit adaptér z 3,3V logiky na 24V. Schéma zapojení tohoto adaptéru je na obr. 5.3.



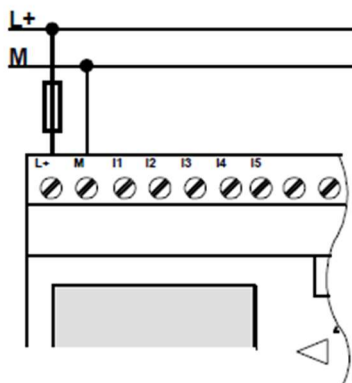
Obr. 5.2 Schéma elektrického zapojení (obrázky zařízení čerpány z katalogů prodejců)



Obr. 5.3 Schéma zapojení adaptéru z 3,3V logiky na 24V logiku

Hlavní modul LOGO! je vybaven 8 vstupy. Obecně jsou všechny vstupy brány jako digitální. Pokud je nutná práce s analogovými vstupy je vyhrazen přesný počet vstupů, které lze zapojit jako analogové. Každý typ modulu LOGO! má odlišný počet těchto konfigurovatelných vstupů. K realizaci projektu byl k dispozici model LOGO! 12/24 RC/RCo, který má rozvržení vstupů následovné: vstupy I1 až I6 jsou vyhrazeny pouze pro digitální vstupy a vstupy I7, I8 lze konfigurovat jako analogové. V tomto případě bylo nutné zapojit dva analogové vstupy pro optoelektronický snímač a teplotní senzor. Potřebným rozšířením je rozšiřující modul LOGO! DM8 24 4i/4o. Tento modul zajišťuje pouze potřebný počet výstupů. Dalším rozšiřujícím modulem je LOGO AM2. Tento modul zajišťuje připojení teplotního senzoru PT1000, který plní pouze ukázkovou funkci a měří okolní teplotu. Jedná se o ukázkou odlišné varianty zapojení teplotních senzorů pro moduly LOGO!. Samotný modul LOGO! nelze propojit přímo se senzorem PT1000, a proto je potřeba zavést tento typ rozšiřujícího modulu.

Moduly LOGO! lze napájet jak stejnosměrným tak střídavým napětím. Existují obě varianty modulů. V našem případě byl modul napájen 24V stejnosměrného napětí. Schéma zapojení je na obr. 5.4. U vybraného rozšiřujícího digitálního modulu (LOGO! DM8 24 4i/4o) bylo důležité pohlídat stejnou třídu napětí, což je podmínkou pro správnou kooperaci s hlavním modulem. Naopak třídu napájení není nutné hlídat u modulu LOGO! AM2, na který třída napětí nemá vliv.

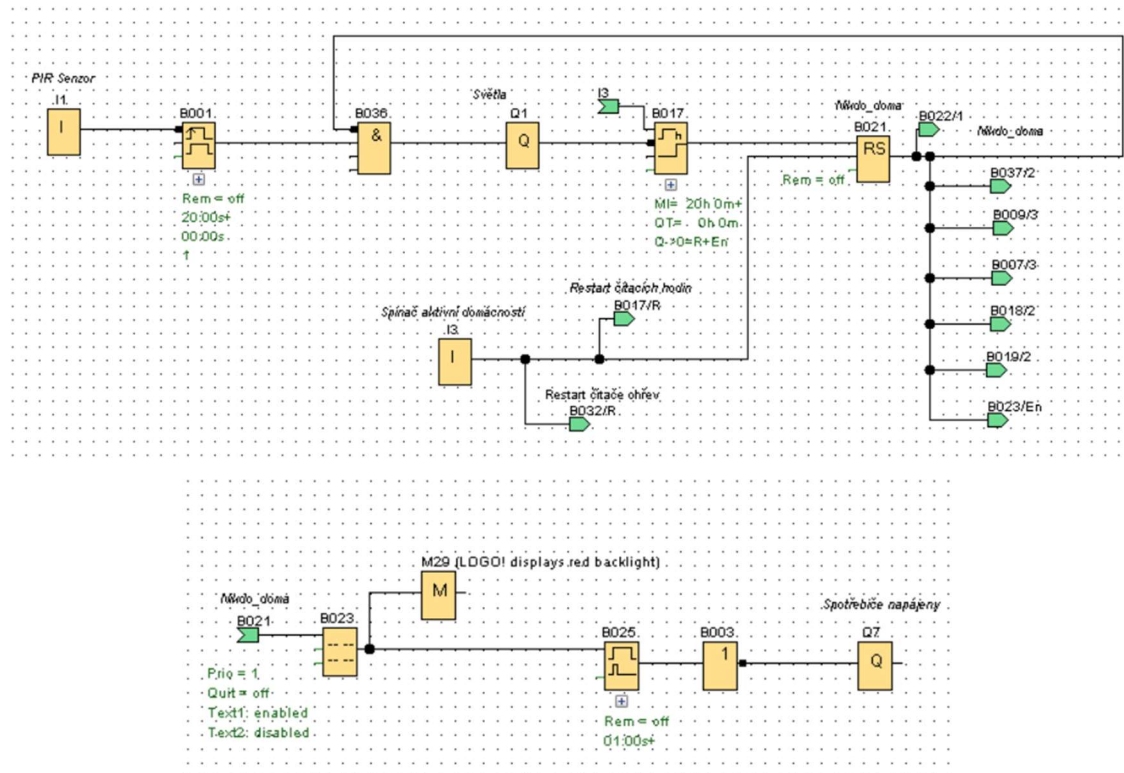


Obr. 5.4 Schéma připojení napájení k modulu LOGO! [7]

5.2 Realizace systému osvětlení a kontrola aktivity v domácnosti

Cílem systému osvětlení je adekvátní reakce na pohyb v místnosti a následná úspora způsobená zkrácením doby svitu žárovky. Na četnost rozsvícení je navázán následný program kontroly aktivity v domácnosti. V modelu domečku jsou pro tyto účely umístěny PIR senzor, 5 malých LED žárovek, plastové modely vytisknuté na 3D tiskárně (stropní osvětlení a zásuvka) a kolébkový spínač. Nejedná se o velikostně srovnatelné modely s reálnými zařízeními.

Samotný program reaguje na digitální vstup I1, který reprezentuje reakci PIR senzoru. Jakmile blokem hranou spouštěné relé (B001) zaznamená vstupní signál, sepne svůj výstup po dobu 20 sekund. Negace na vstupu tohoto bloku je zavedena z důvodu invertovaného výstupu na dostupném PIR senzoru. Volba bloku hranou spouštěné relé byla nutná také kvůli použitému PIR senzoru, kdy při opakovaném snímání docházelo k chvilkové neaktivitě. Zmíněný blok dodá na výstup signál po dostatečně dlouhou dobu, aby eliminoval chybu snímače. Spuštění je podmíněno aktivitou v domácnosti. Kontrolu aktivity zajišťuje prvek čítače provozních hodin (B017). Pokaždé, kdy je rozsvíceno světlo, je čítač resetován na původní nastavenou hodnotu 20 hodin. Pokud během této doby není zaznamenána aktivita, domácnost přechází do pasivního režimu. Jakmile přejde domácnost do pasivního režimu, některé spotřebiče se vypnou a nedochází ke spotřebě energie (výstup Q7 není aktivní). Opětovnou aktivaci provedeme přepnutím spínače I3, který vrací vše do původního stavu. Funkční blok textové zprávy plní pouze informativní funkci pro uživatele, který je informován o stavu domácnosti.

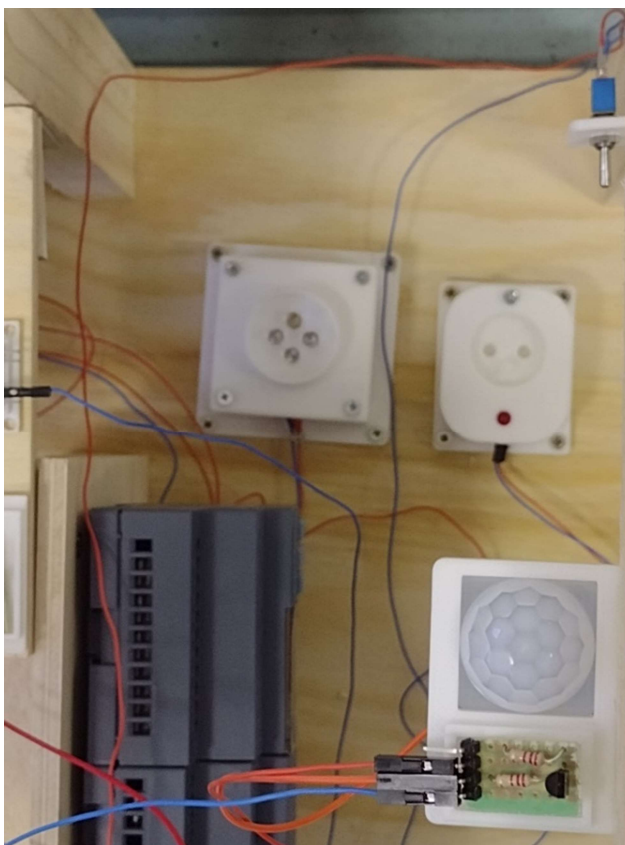


Obr. 5.5 Program osvětlení a kontroly aktivity v domácnosti v LOGO! Soft Comfort

Malé držáky pro PIR senzor a spínač v samotné realizaci byly vytisknuty na 3D tiskárně. Modely stropního osvětlení a zásuvky byly vyrobeny stejnou metodou. Volba malých LED žárovek o průměru 5,3 mm je opodstatněna větším prostorem v modelu pro kabeláž a lepší manipulaci při testování. Vyhloubený prostor uprostřed osvětlení je nutný pro umístění rezistorů, které jsou propojeny s LED žárovkami. Svit z těchto žárovek není tak silný, a proto bylo nutné pro lepší viditelnost umístit čtyři žárovky. Umístění spínače nemá speciální opodstatnění, hlavním cílem bylo pouze ochránění spínače před zničením. Model zásuvky je vybaven znovu malou LED žárovkou, která reprezentuje přívod elektrického proudu k spotřebičům. Pokud je rozsvícena, je možné zapojené spotřebiče používat v běžném režimu. Jakmile přestane svítit, domácnost se nachází v pasivním režimu a není nutné dále spotřebiče napájet.

5.2.1 PIR senzor

Zkratka PIR je odvozena z anglického názvu „passive infrared detector“, neboli pasivní infračervený detektor. Základní princip funkce PIR senzorů je založen na pyroelektrickém jevu. Obecně lze pyroelektrický jev definovat jako schopnost materiálu generovat dočasný elektrický potenciál při změně jeho teploty [19]. Sensory jsou dále složeny z PIR elementu, optiky a procesoru, který zpracovává výstup. Optika PIR senzorů je soustředěna na rozeznání infračerveného záření vyzařující z povrchu objektů v detekčních zónách. V modelu domečku je umístěn senzor, který dokáže sledovat oblast až ve vzdálenosti 6 metrů a v poli 110°. Základním prvkem PIR senzoru je pyroelement. Jedná se o polovodičovou součástku (ze sloučenin na bázi lithia a tantalu), která je citlivá na ozáření infračerveným světlem tak, že začne generovat elektrický povrchový náboj Q [19]. Pokud je změna ve velikosti dopadajícího záření, tak dojde ke změně elektrického povrchového náboje Q . Dalším krokem je samotné zpracování signálu. V našem případě se jedná o digitální PIR senzor, takže je nutné převést analogový signál na digitální. Převod je prováděn prvotním navzorkováním analogového signálu a následně je použito kvantování tohoto signálu. Pro kvalitnější výsledek jsou dále použity filtry pro vyhlazení výsledného signálu. Závěrečné relé, obecně spínač, reprezentuje výstup PIR senzoru. Důležitou roli ve správném fungování PIR senzorů hraje pozice. V modelu domečku bylo důležité umístit senzor, tak aby optika nesměřovala k zařízením. Pokud bychom nasměrovali senzor k LED žárovkám, tak by mohlo docházet k chybným reakcím z důvodu infračerveného záření vyzařujícího z žárovek. Proto byl umístěn směrem vzhůru, aby reagoval pouze na pohyb nad domečkem.



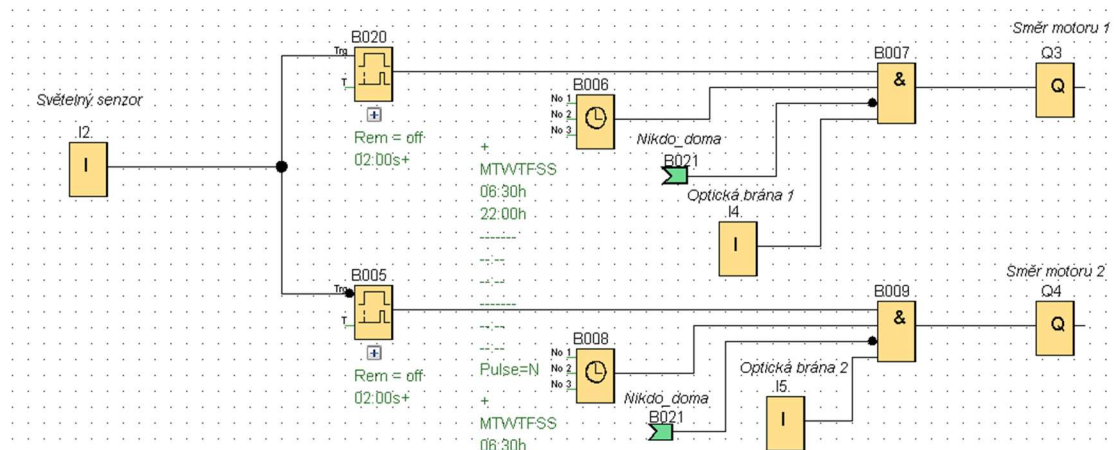
Obr. 5.6 Realizace osvětlení a kontroly aktivity v domácnosti

5.3 Realizace automatizované rolety

Automatizované rolety řeší jak problém úspor energie, tak i komfort bydlení. Využití většího množství denního světla napomáhá k méně častému používání osvětlení a částečně může napomoci k rychlejšímu vytopení místnosti. K vyhotovení systému rolety byly použity malé součástky vyrobené na 3D tiskárně jako kladka, držáky zařízení apod. Dále bylo nutné zajistit pohon rolety. O pohon se postaral malý stejnosměrný motorek, na který lze narazit např. v modelech aut na ovládání. Pro změnu směru otáčení motoru je nutné propojení s H-můstkem. O vymezení prostoru, v jakém se roleta může pohybovat, se starají dva optické mikrosplínače (optické brány). Posledním zařízením v tomto systému je malý světelný senzor.

Základem programu automatizované rolety jsou digitální vstup I2 (světelný senzor) a týdenní spínací hodiny (B006 a B008). Pokud není vstup I2 aktivní, senzor zaznamenává dostatečné světlo. V tomto případě je roleta vytažena do horní polohy (směr motoru 2). Naopak pokud je vstup aktivní, světelný senzor zaznamenal tmu a roleta je spuštěna do spodní polohy. O zastavení pohybu se starají dva digitální vstupy I4 a I5. Jde o optické brány, které zamezí otáčení motorku. Čas, kdy je automatizovaná roleta aktivní, je nastaven ve spínacích hodinách. V tomto případě požadujeme aktivitu rolety v časech od 6:30 do 22:00. Nehrozí tedy, že by o půlnoci nebo v brzkých ranních hodinách roleta

konala pohyb. Celá aktivita je podmíněna obýváním domácnosti (bloky s názvem Nikdo_doma).



Obr. 5.7 Program automatizovaných rolet v LOGO! Soft Comfort

Samotná realizace spočívá ve vytahování, popřípadě spouštění, malého kvádru ve vymezeném prostoru dvěma optickými branami. Jakmile spínač zaznamená pohyb, dojde k zastavení motorku. Všechna zařízení jsou upevněna k dřevěnému modelu pomocí držáků, které byly navrženy a následně vytisknuty na 3D tiskárně. Pro testování funkčnosti světelného senzoru není nutné čekat na venkovní setmění. Zakrytím optiky senzoru rukou docílíme dostatečné tmy a dojde k sepnutí relé. Provozní doba automatizované rolety je kontrolována v modulu LOGO!.

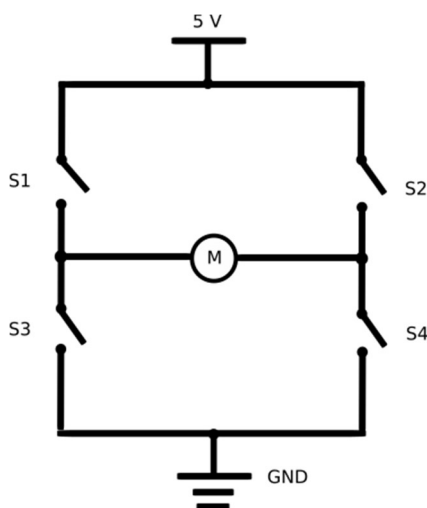
5.3.1 Světelný senzor

Zvolený světelný senzor je napájen stejnosměrným napětí 5 V. Jde o malý senzor reagující na intenzitu okolního světla. Relé sepne pokud intenzita světla překročí nastavenou hodnotu. Citlivost sepnutí lze i v těchto zařízeních měnit. Základní princip fungování světelných senzorů spočívá v rozpoznávání stupně intenzity světla díky fotobuňkám. V některých případech lze nalézt jiné označení např. fotorezistor nebo fototranzistor. Ve tmě klesá intenzita světelného toku a tím se mění parametry fotobuňky. Dochází ke změně elektrické vodivosti a následnému sepnutí relé [20]. Obecně v domácnostech lze využít i jiná zařízení jako např. soumrakový spínač, ale výhodou světelných senzorů je právě nastavitelná citlivost reakce.

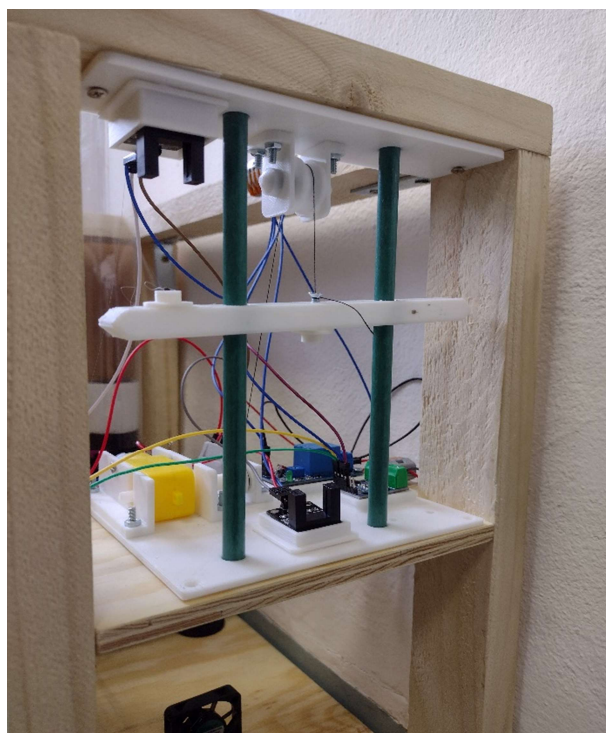
5.3.2 Řízení stejnosměrného motoru H můstkem

Hlavním problémem v realizaci systému rolety bylo dosažení opačného směru otáčení DC motorku. Stejnosměrný motor je točivý stroj napájený stejnosměrným proudem, kde přepólováním napájecího napětí docílíme točení motoru na opačnou stranu [21]. Ke změně polarity napájecího napětí využíváme H můstek. H můstek je elektrický obvod sestavený z tranzistorů určený pro řízení stejnosměrných motorů [21]. Tento obvod obsahuje čtyři spínací prvky, které jsou zapojeny do tvaru písmene H. Mezi těmito prvky je zapojeno zařízení, s kterým chceme pracovat. Schéma zapojení je zobrazeno na Obr.

5.8. Podle toho, které spínače jsou sepnuté a které rozepnuté, můžeme dostat tři stavy – rozpojený obvod, průchod proudem zařízením v jednom směru a průchod zařízením v druhém směru [22]. V praxi pokud jsou sepnuty spínače S1 a S4 motor se otáčí v jednom směru, pro změnu směru otáčení je naopak nutné sepnout pouze spínače S2 a S3. V systému rolety tedy propojíme H můstek s modulem LOGO! a následně H můstek s DC motorkem. Na vstupy H můstku přivádíme výstupy Q2 a Q3 modulu LOGO!, které inicializují změnu směru otáčení motorku. Vybraný H můstek je napájen 5V stejnosměrného napětí.



Obr. 5.8 Schéma zapojení H můstku s motorem

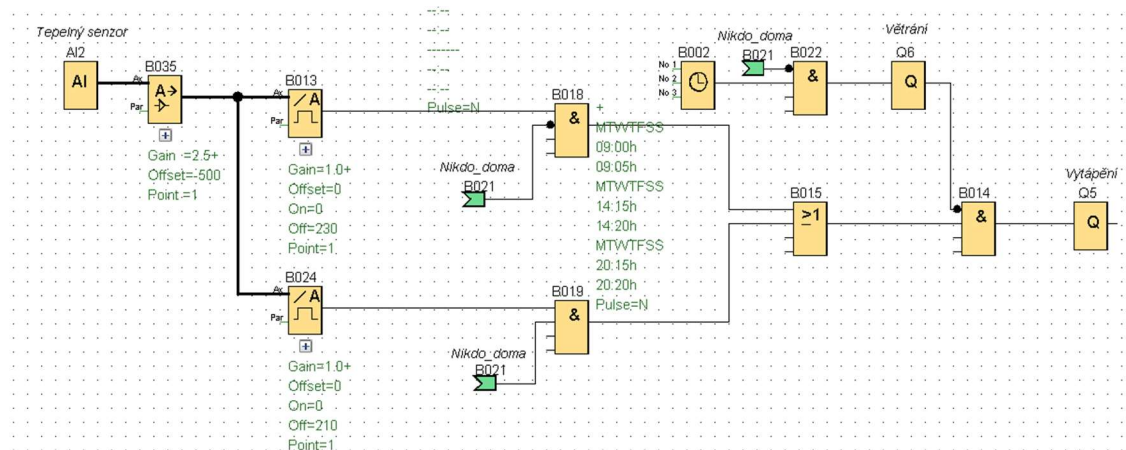


Obr. 5.9 Realizace automatizované rolety

5.4 Realizace vytápění místnosti

Systém vytápění místnosti je navržen pro automatizovanou regulaci teploty. Cílem je udržovatelnost nastavené hodnoty teploty pro větší komfort bydlení a úsporu energie. Tento systém bude reprezentován v modelu plastovou krabičkou, ve které budou umístěny rezistor, tepelný senzor a malý větráček pro chlazení PC. Větráček simuluje pravidelné větrání v místnosti.

V programu jsou realizovány dvě varianty vytápění. Horní větev programu reprezentuje vytápění pro aktivní domácnost. Sledování hodnoty teplotního senzoru je prováděno v bloku analogový spínač (B013), kde je nastaveno sepnutí výstupu pro hodnoty menší než 230. V rámci nastavení bloku bylo nastaveno sledování hodnot s jedním desetinným místem. Hodnota 230 je ve skutečnosti 23,0. Pro práci s teplotami je tato varianta komfortnější. Stejný princip sledování je nastaven u spodní větve, která reprezentuje vytápění v pasivní domácnosti. Rozhodnutí jestli vytápíme aktivní nebo pasivní domácnost provádíme v blocích AND (B018 a B019). Pokud je domácnost aktivní, blok Nikdo_doma (B021) není sepnut. Pro správnou funkčnost je nutné na tento vstup zavést negaci (reprezentována černou tečkou na vstupu). Opačný případ je na prvku AND pro pasivní domácnost. Zde je naopak očekáváno sepnutí prvku Nikdo_doma. Pravidelné intenzivní větrání lze měnit v bloku týdenních spínacích hodin (B002). Důležitou součástí systému je zamezení vytápění v případě větrání. Pokud je aktivní větrání nelze vytápět. Tato podmínka je realizována znovu v bloku AND (B014). Funkčnost větrání je podmíněna aktivitou v domácnosti.



Obr. 5.10 Program vytápění místnosti v LOGO! Soft Comfort

V rámci modelu domečku je vytápění řešeno kompaktní krabičkou, která obsahuje všechna zařízení. Zařízení jsou propojena na desce plošných spojů. Dostupný větráček v modelu je využíván obvykle jako chlazení uvnitř skříní počítačů. Napájen je napětím 12 V. Jako zdroj tepla byl zvolen malý rezistor o odporu 150 Ω . Pro tento typ malé krabičky postačil jeden rezistor, ke kterému byl přímo přiložen tepelný senzor. Tato varianta řešení je výhodnější pro lepší názornost řešení. V rámci větších prostorů by bylo

nutné použít více rezistorů. Pro přibližný výpočet potřebného tepelného výkonu rezistorů, lze použít vzorec pro výpočet tepelné kapacity $C [J * K^{-1}]$

$$C = \sum c_i m_i = \sum c_i \rho_i V_i , \quad (18)$$

kde c je měrná tepelná kapacita, ρ hustota objektu a V objem objektu.

Po získání hodnoty tepelné kapacity jsme schopni zjistit tepelný výkon. Podle výsledného tepelného výkonu jsme dále schopni přibližně rozhodnout o počtu potřebných rezistorů na vytopení prostoru. Důležitou součástí výpočtu je správný převod jednotek. Vzorec pro výpočet tepelného výkonu $P [W]$:

$$P = \frac{C * \Delta T}{t} , \quad (19)$$

kde ΔT je teplotní rozdíl, o který se změní teplota a t doba, za kterou ke změně dojde.

Pro ukázkou řešení vzorového příkladu je přiložen výpočet z prostředí Matlab. Jde o výpočet tepelného výkonu pro vytopení veškeré hmoty nacházející se ve vybraném prostoru. Ve výpočtu si lze zvolit dostupný počet rezistorů, požadovaný teplotní rozdíl a dobu, za kterou dojde k této změně. Jedná se o přibližný výpočet, který slouží pro přibližnou představu o počtu potřebných rezistorů. Výpočet je zaměřen pro dřevěný kvádr o rozměrech 165x150x120 mm s plastovou zábranou z jedné strany. Jedná se o možnou variantu, která by mohla být součástí skutečného modelu domečku. Počet používaných rezistorů byl zvolen 15, teplotní rozdíl 10 K a doba za kterou dojde ke změně 90 sekund. Potřebný tepelný výkon P vyšel přibližně 68 W. Na jeden rezistor je možné přivést maximálně 5 W. Z výsledku vyplývá že 15 rezistorů by pro tuto situaci byl dostatečný počet, ale kvůli velkému množství použitých rezistorů není tak efektivní pro tento typ modelu. Proto byla vybrána varianta s jedním rezistorem a přiloženým tepelným senzorem v menší kompaktní krabičce. Jde o výrazně názornější, rychlejší a méně nákladnou variantu ukázkou vytápění.

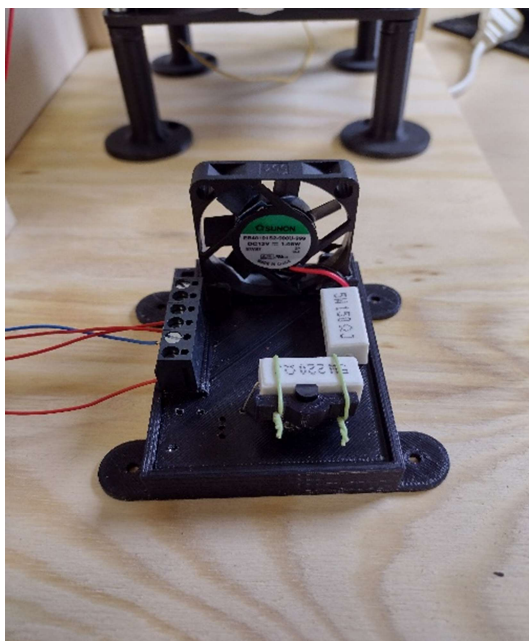
```

hust_dreva      = 400; %kg/m3
hust_plast      = 1300; %kg/m3
hust_vzduchu    = 1.275;%kg/m3

V_sloup1        = 0.000108; %m3
V_sloup2        = 0.000027; %m3
V_sloup3        = 0.000054; %m3
V_steny         = 0.000144;%m3
V_plastzabrana  = 0.000061;%m3
V_rezistor      = 0.0000019;%m3
V_prostoru      = 0.165*0.150*0.120;%m3
pocetR          = 15;
m_rezistor      = 0.005;%kg
c_keramika      = 840;%Jkg-1K-1
c_drevo         = 1500;%Jkg-1K-1
c_plast         = 1100;%Jkg-1K-1
c_vzduchu       = 1005;%Jkg-1K-1
delta_T         = 10;%K
t               = 90;%s
C = (c_keramika*m_rezistor*pocetR)+ ...
(c_drevo*(V_sloup1+V_sloup2+V_sloup3+4*V_steny)*hust_dreva)+...
(c_plast*V_plastzabrana*hust_plast)+(c_vzduchu * hust_vzduchu * V_prostoru)
P = C*delta_T/t
C = 613.0357
P = 68.1151

```

Obr. 5.11 Ukázkový příklad výpočtu tepelného výkonu v prostředí Matlab

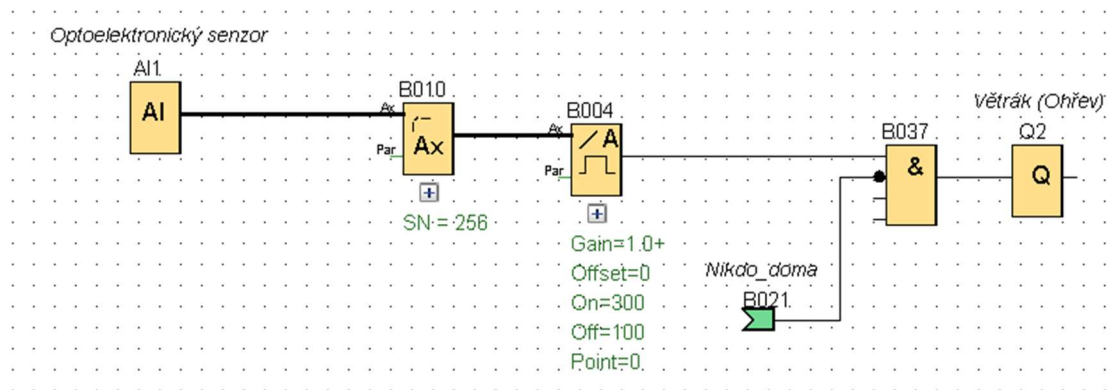


Obr. 5.12 Realizace vytápění místnosti v modelu domečku

5.5 Realizace regulace teploty vody v bojleru

System je navržen pro automatizované řízení teploty vody v chytrých bojlerech. V současných moderních domácnostech se setkáváme s těmito zařízení. Samotný řídicí program regulace teploty je možné využít ve více aplikacích, kde sledujeme určitou hranici. V rámci této realizace byl zvolen odlišný přístup řešení než při sledování teploty v místnosti. Důvodem odlišné volby řešení byla dostupnost zařízení, která byla k dispozici a vyzkoušení jiné varianty programu pro sledování hraniční hodnoty. Hlavní myšlenka tohoto řešení spočívá v řízení zařízení pro vzduchovou levitaci pomocí modulu LOGO!. Sledovaným předmětem bude míček v průhledné trubici, který neklesne pod zadanou hranici. Jde o stejný princip jako v bojlerech, kde není povoleno klesnout pod určitou hranici teploty vody.

Hlavním blokem samotného programu je blok analogového spínače (B004). Hraniční hodnoty v tomto bloku jsou nastaveny na 300 a 100. Tyto hodnoty nereprezentují skutečnou vzdálenost tělesa od optoelektronického senzoru. Jde o vnitřní hodnoty senzoru, které vysílá do modulu LOGO!. Pro zjištění těchto hodnot bylo využito testování skutečného modelu. Předcházející blok analogový filtr (B010) pouze vyhlazuje signál od možného šumu. V nastavení bloku jsou různé velikosti vyhlazení. Zvolená hodnota SN 256 je nejvyšší možná. Celý program zajišťuje pohyb válečku v trubici a funkčnost je znovu podmíněna aktivitou v domácnosti.

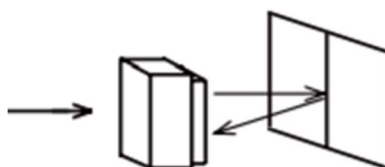


Obr. 5.13 Program regulace teploty vody v bojleru v LOGO! Soft Comfort

K samotné realizaci v modelu domečku bylo použito zařízení pro vzduchovou levitaci a optoelektronický senzor od firmy Sharp. Kvůli větší velikosti zařízení bylo nutné vymezit dostatečný prostor v modelu. Optoelektronický snímač byl vložen do spodní části zařízení pro levitaci, tak aby dokázal snímat kompletní trubici zařízení. Napájení optoelektronického snímače je 5V. Zařízení pro vzduchovou levitaci je složeno z následujících komponent: ventilátor, plastové podpurné nohy, držák průhledné trubice a samotná trubice.

5.5.1 Optoelektronický snímač

Optoelektronické snímače jsou založeny na šíření elektromagnetických vln (světelných paprsků) [23]. Výhoda těchto snímačů spočívá ve velmi dobré schopnosti odolávat vůči rušení elektromagnetickým. Dále tyto snímače velmi dobře reagují v prostředí s vysokou vlhkostí a prašností. Optoelektronické snímače jsou schopny měřit ve velkém rozsahu vzdáleností a detekovat objekt v delších vzdálenostech. Další jejich výhodou je velká reakční rychlost, kdy lze měření opakovat i několikrát za sekundu [23]. Existuje velké množství typů optoelektronických snímačů např. absolutní snímače nebo inkrementální snímače. Zjednodušený princip fungování těchto snímačů spočívá ve vysílání světelných paprsků z vlastního zdroje (např. LED diody nebo laserové diody) a následném příjmu odražených paprsků od objektu. Následně se optický signál převádí na elektrický pomocí fotodiod nebo fototranzistorů [24]. Tato zařízení nejčastěji najdou využití v místech, kde je třeba hlídat potřebné rozměry, hladinu nebo obecně jako světelné závory.



Obr. 5.14 Zjednodušené schéma principu měření optoelektronického snímače [25]



Obr. 5.15 Realizace regulace teploty vody v bojleru

6 ZÁVĚR

Kompaktní modul LOGO! od firmy Siemens nabízí velmi zajímavou alternativu pro řízení chytrých domácností v oblasti programovatelných automatů. Výsledkem práce byl vyrobený model domečku obsahující modul LOGO! a zařízení reprezentující chytrou domácnost. Sestavený model může dále sloužit k edukativním účelům na Ústavu automatizace a informatiky.

Teoretická část práce byla věnována logickému řízení a popisu použitého programovacího prostředí LOGO! Soft Comfort. Praktická část byla věnována zjednodušenému návrhu bytové jednotky a systémů zajišťující úsporu. Jednalo se o navržení bytové jednotky a následném popisu automatizovaných systémů. Pro jednotlivé systémy byla vybrána zařízení, která by bylo možné použít pro samotnou realizaci. Veškeré návrhy byly cíleny na energetickou úsporu v domácnosti. Tento požadavek je podložen procentuálními úsporami pro jednotlivé systémy. Celkové roční úspory, v poplatcích za elektrickou energii, se mohou pohybovat v rozmezí 5500 až 6500 Kč. Jedná se o přibližný přepočít, který ukazuje pouze na možnou úsporu zavedením automatizovaných systémů. Dále je praktická část zaměřena na samotnou realizaci modelu domečku. Jedná se o dřevěný model o velikosti 450x400x350 mm. Model je rozdělen do několika částí, kde jsou umístěny jednotlivé systémy zmíněné v návrhu bytové jednotky. Nejde o velikostně srovnatelné modely zařízení se skutečností, ale funkcionality je totožná jako ve skutečných domácnostech. Řídící programy byly naprogramované v prostředí LOGO! Soft Comfort. V samotném textu jsou uvedeny obrázky řídicích programů a fotografie částí fyzického modelu. Systémy pro vypnutí vody a plynu byly po domluvě vynechány, z důvodu nespelnitelných požadavků na bezpečnou manipulaci a ochranu ostatních zařízení vůči kapalině.

Práci považuji za přínosnou díky uvedení do problematiky programovatelných automatů. Vyzkoušel jsem si elektrická zapojení jednotlivých zařízení, práci s jazykem funkčních bloků v rámci prostředí LOGO! Soft Comfort, prostředí Matlab a modelování v softwaru Autodesk Inventor. Tyto zkušenosti s kompletním návrhem domácnosti lze využít při vlastní přestavbě domácnosti.

7 SEZNAM POUŽITÉ LITERATURY

- [1] ŠVARC Ivan, MATOUŠEK Radomil, ŠEDA Miloš a VÍTEČKOVÁ Miluše. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4398-3.
- [2] MIEKISCH, Jiří. *Booleova algebra* [online]. Střední škola průmyslová a umělecká, Opava, příspěvková organizace, Praskova 399/8, Opava, 746 01, 2011 [cit.2021-01-18].
Dostupné z: https://www.sspuopava.cz/static/UserFiles/File/sablony/AUT_IV/VY_32_INOVACE_E-15-07.pdf. Inovace a zkvalitnění výuky prostřednictvím ICT (20 vzdělávacích materiálů). Střední škola průmyslová a umělecká, Opava, Praskova 399/8, Opava, 746 01.
- [3] JANČÍK, David. Hradla, klopné obvody RS, Polovodičové paměti. *Maturitní otázky z Technického vybavení počítače* [online]. itnetwork.cz, 2013 [cit. 2021-01-18].
Dostupné z: <https://www.itnetwork.cz/maturitni-otazky-technicke-vybaveni-pocitace-hradla-klopne-obvody-rs-a-polovodicove-pameti>
- [4] ISŠT, učební materiály. *PA - PLC, programové logické automaty* [online]. 2015 [cit. 2021-01-18]. Dostupné z: <http://isst.hys.cz/images/prezentace/PA-22.pdf>
- [5] Automa.cz. *Programovatelné automaty – PLC, nebo PAC?* [online]. Kolín: Teco a.s, 2003 [cit. 2021-01-18].
Dostupné z: https://automa.cz/cz/casopis-clanky/programovatelne-automaty-plc-nebo-pac-2003_06_28832_683/
- [6] ŠERÝ, Michal. *Teorie systémů*. [online prezentace]. 2019 [cit. 2021-02-26]. Dostupné z: http://home.pf.jcu.cz/~kyklop/SERYM/automatizace/pred/pdf/Automatizace_systemy.pdf
- [7] SIEMENS LOGO! *Manual. Industry Support Siemens* [on-line]. Munich: Siemens, ©2017 [cit. 2021-02-28].
Dostupné z: https://cache.industry.siemens.com/dl/files/041/109741041/att_924628/v1/logo_system_manual_en-US_en-US.pdf<https://it-slovník.cz/pojem/drag-and-drop>
- [8] LOGO! Software. *New.siemens.com* [online]. [cit. 2021-02-28]. Dostupné z: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-o-software.html#LOGOSoftComfort>
- [9] ITSlovník.cz [online]. [cit.2021-02-26]. Dostupné z :<https://it-slovník.cz/pojem/drag-and-drop>
- [10] mylms.cz [online]. 2019 [cit.2021-02-26]. Dostupné z: <https://www.mylms.cz/programovani-siemens-logo-1-o-co-vlastne-jde/>
- [11] WAGNEROVÁ, Renata a Jiří TŮMA. *Základy automatizace* [online]. Ostrava: Vysoká škola báňská - Technická univerzita, [2008] [cit. 2021-03-15]. ISBN isbn978-80-248-1523-7. Dostupné z: http://www.elearn.vsb.cz/archivcd/FS/Zaut/Skripta_text.pdf
- [12] Elektrina.cz [online]. [cit. 2021-5-6]. Dostupné z: <https://www.elektrina.cz/cena-elektriny-za-kwh-2020-cez-eon-pre-bohemia-centropol-a-dalsit-led.cz>

- [13] led.cz [online]. [cit. 2021-03-18].
Dostupné z: <https://www.t-led.cz/blog/pohybova-cidla-senzory/>
- [14] novinky.cz [online]. [cit. 2021-03-18].
Dostupné z: <https://www.novinky.cz/bydleni/tipy-a-trendy/clanek/podlahove-topeni-je-stale-oblibenejsi-ma-ale-i-sva-uskali-40044410>
- [15] GRYGERA, Filip a Alice KUPČEKOVÁ. *Bydlete úsporně: jak investovat do energetických úspor a získat dotaci v programu Zelená úsporám*. Brno: Computer Press, 2010. ISBN 978-80-251-2857-2.
- [16] Bydleni.cz [online]. [cit. 2021-5-6]. Dostupné z: <https://www.bydleni.cz/clanek/7x-USPORY-chytrou-regulaci-tepla>
- [17] eon.cz [online]. [cit. 2021-03-18]. Dostupné z: <https://www.eon.cz/radce/uspورا-energie/uspory-elektriny-a-plynu/co-je-chytry-bojler-a-co-umi>
- [18] ceskykutil.cz [online]. [cit. 2021-05-1]. Dostupné z: <https://ceskykutil.cz/clanek-13107-chytry-bojler-vam-muze-use-trit-tisice>
- [19] vyvoj.hw.cz [online]. [cit. 2021-04-29]. Dostupné z: <https://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>
- [20] myhouse.technoluxpro.com [online]. [cit. 2021-05-1]. Dostupné z: <https://myhouse.technoluxpro.com/cs/elektrosnabzhenie/kak-podklyuchit-ulichnyj-datchik-osveshheniya-dlya-vklyucheniya-sveta/>
- [21] PAŠTYKA, Jan. *DC motory řízené pomocí H můstku* [online]. 2017 [cit. 2021-05-1]. Dostupné z: https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H_Bridge.pdf. Dokumentace. České vysoké učení technické v Praze Fakulta elektrotechnická.
- [22] navody.drateg.cz [online]. [cit. 2021-05-2]. Dostupné z: <https://navody.drateg.cz/technikuv-blog/zmena-smeru-otaceni-dc-motoru.html>
- [23] STRAKA, Radomír. *Návrh optoelektronického snímače polohy* [online]. Brno, 2010 [cit. 2021-5-2]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=27785. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Jan Puskely.
- [24] euc.kr-olomoucky.cz [online]. [cit. 2021-5-3]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/960>
- [25] farnell.com [online]. 2006 [cit. 2021-5-4]. Dostupné z: <http://www.farnell.com/datasheets/1657845.pdf>

8 SEZNAM PŘÍLOH

1. Složka ProgramLOGO obsahující kompletní řídicí program vytvořený v prostředí LOGO! Soft Comfort a soubor pdf, kde je program kompletně vyexportován.
2. Fotografie kompletního modelu domečku.

PŘÍLOHY

2. Fotografie kompletního modelu.

