

# Model inteligentního domu řízený mikropočítačem

Bc. Tomáš Kyjania

---

Diplomová práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Kyjania**  
Osobní číslo: **A12391**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **prezenční**

Téma práce: **Model inteligentního domu řízený mikropočítačem**

## Zásady pro vypracování:

1. Navrhněte funkce pro model domku vhodné pro demonstraci možností ovládání a regulace v moderním domě.
2. Navrhněte vhodné rozhraní pro ovládání těchto funkcí s možností snadné úpravy ovládaných funkcí uživatelem modelu.
3. Vytvořte model domu včetně rozhraní pro jeho ovládání.
4. Vytvořte programové vybavení pro mikropočítač použitý pro řízení modelu domku.
5. Vytvořte programové vybavení pro osobní počítač nebo jiné zařízení vhodné pro uživatelské úpravy ovládaných funkcí domku.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0.**
2. **CATSOULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3.**
3. **MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMELE AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.**
4. **PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-730-0110-1.**
5. **VALEŠ, Miroslav. Inteligentní dům. 1. vyd. Brno: ERA, 123 s., il. (část barev.). ISBN 80-736-6062-8.**

Vedoucí diplomové práce:

**Ing. Jan Dolinay, Ph.D.**

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

**21. února 2014**

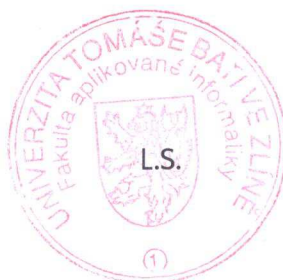
Termín odevzdání diplomové práce:

**20. května 2014**

Ve Zlíně dne 21. února 2014



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Pojem inteligentní budova je v moderní době velmi aktuální. Cílem této práce je podrobně se seznámit se souvisejícími pojmy a vytvořit funkční model inteligentního domu, jak po hardwarové stránce, tak po softwarové stránce. Model bude sloužit jako exponát na výstavu „Experimentarium“ v Otrokovicích, a jeho účelem je navést studenty základní školy na IT obor. Pro ovládání modelu byla vytvořena řídicí aplikace pro PC v jazyce C++ s knihovnou wxWidgets. Model je řízen mikropočítačem, pro nějž byl vytvořen software ve vývojovém prostředí Arduino.

Klíčová slova: inteligentní dům, mikropočítač, C++, Arduino

## **ABSTRACT**

Term intelligent building is very actual nowadays. Goal of this thesis is to become acquainted with relevant terms and to build functional model of an intelligent building. The model will be exposed on „Experimentarium“ exposition in Otrokovice city. Purpose of the model is to lead elementary school students to choose IT specialization. Controlling application in C++ language with wxWidgets was created to control the model from PC. Model is controlled by a microcontroller, software for this microcontroller was created in Arduino environment.

Keywords: intelligent building, microcontroller, C++, Arduino

Poděkování:

Chtěl bych poděkovat především panu Ing. Janu Dolinayovi, Ph.D. za pomoc a rady, které mi byly cennou oporou při zpracování mé práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

## OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 INTELIGENTNÍ BUDOVA</b> .....	<b>11</b>
1.1.1 Definice č. 1 .....	11
1.1.2 Definice č. 2 .....	11
1.1.3 Definice č. 3 .....	11
<b>2 INTELIGENTNÍ DŮM</b> .....	<b>12</b>
2.1 TYPICKÝ INTELIGENTNÍ DŮM .....	12
2.2 INTELIGENTNÍ ELEKTROINSTALACE.....	12
<b>3 POUŽITÉ SOFTWAREVÉ TECHNOLOGIE</b> .....	<b>14</b>
3.1 PROGRAMOVACÍ JAZYK C++ .....	14
3.1.1 Objektově orientované programování.....	14
3.2 KNIHOVNA WXWIDGETS .....	14
3.3 VÝVOJOVÉ PROSTŘEDÍ ARDUINO.....	15
3.4 EAGLE.....	17
3.4.1 Editor schémat.....	17
3.4.2 Editor spojů .....	17
3.4.3 Autorouter .....	18
<b>4 VÝVOJOVÝ KIT ARDUINO</b> .....	<b>19</b>
<b>II PRAKTICKÁ ČÁST</b> .....	<b>21</b>
<b>5 NÁVRH MODELU</b> .....	<b>22</b>
5.1 FUNKČNOST MODELU .....	22
5.1.1 Manuální ovládání.....	23
5.1.2 Automatické ovládání .....	23
5.2 FYZICKÝ MODEL DOMU .....	24
5.2.1 Použité materiály.....	24
5.2.2 Technický návrh modelu .....	24
<b>6 ELEKTROINSTALACE V MODELU</b> .....	<b>27</b>
6.1 AKČNÍ PRVKY .....	27
6.1.1 Světla.....	28
6.1.2 Ventilátor.....	28
6.1.3 Topení .....	29
6.1.4 Roletka .....	29
6.2 SNÍMAČE .....	30
6.2.1 Teplotní snímače .....	30
6.2.2 Pohybový snímač .....	31
6.2.3 Infračervený snímač .....	32
6.3 PLOŠNÝ SPOJ SE SPÍNACÍ ELEKTRONIKOU.....	32
6.3.1 Spínání světel .....	32

6.3.2	Spínání ventilátoru .....	33
6.3.3	Spínání topení.....	34
6.3.4	Spínání roletky .....	35
6.3.5	Návrh kombinačního obvodu .....	37
6.3.6	Zapouzdření spínací elektroniky .....	39
<b>7</b>	<b>VÝROBA.....</b>	<b>41</b>
7.1	VÝROBA MODELU DOMU .....	41
7.2	VYHOTOVENÍ DESKY S PLOŠNÝM SPOJEM .....	41
7.2.1	Návrh desky.....	42
7.2.2	Výroba desky.....	44
7.2.3	Osazení desky součástkami .....	45
7.2.4	Testování desky.....	45
7.3	NAPÁJENÍ .....	46
7.3.1	Napájení 12V.....	46
7.3.2	Napájení 5V.....	46
7.4	ELEKTROINSTALAČNÍ MATERIÁL .....	47
7.4.1	Kabeláž.....	47
7.4.2	Další použitý materiál .....	47
<b>8</b>	<b>SOFTWARE .....</b>	<b>48</b>
8.1	ŘÍDÍCÍ APLIKACE NA PC .....	48
8.1.1	Vzhled řídicí aplikace.....	48
8.1.2	Třída Prvek.....	49
8.1.3	Sériová komunikace .....	50
8.2	ŘÍDÍCÍ SOFTWARE PRO MIKROPOČÍTAČ .....	50
8.2.1	Třída snímač.....	51
8.2.2	Třída Prvek.....	51
8.2.3	Hlavní smyčka programu .....	53
8.2.4	Časování .....	54
	<b>ZÁVĚR .....</b>	<b>55</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>56</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>57</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>58</b>
	<b>SEZNAM TABULEK.....</b>	<b>60</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>61</b>

## ÚVOD

Inteligentní domy začínají být v současnosti velmi žádané. Přičemž za onu inteligenci se obecně považuje schopnost domu přiměřeně reagovat na změny prostředí, a to za účelem vytvářet v domě příznivé podmínky a pohodu pro člověka. S rozvojem moderních technologií jsou tak inteligentní domy stále častějším jevem. [5]

Tato práce je rozdělena na teoretickou a praktickou část. Teoretická část se věnuje teorii o inteligentních budovách a inteligentních domech, včetně několika definic těchto pojmů. Dále jsou zde teoreticky popsány všechny použité softwarové a hardwarové technologie, jako jsou C++, wxWidgets, EAGLE a vývojová platforma Arduino.

Cílem praktické části práce je navrhnout a vyrobit funkční model inteligentního domu. Jsou zde navrženy funkce modelu tak, aby byl model vhodný pro demonstraci možností ovládání a regulace v moderním domě.

Dále je navrženo a popsáno rozhraní pro úpravu ovládacích funkcí, které je realizováno pomocí řídicího osobního počítače připojeného k modelu USB kabelem. Pro tento počítač je navržen a vytvořen řídicí software v jazyce C++ s využitím knihovny wxWidgets.

Dále je v praktické části vyroben dřevěný model domu, který plní zadané a navržené funkce. Model je řízen mikropočítačem, a to s využitím vývojové platformy Arduino Uno. Pro model je navržena a vyhotovena veškerá elektronika, realizována na desce plošného spoje. Tento plošný spoj byl navržen a vyroben svépomocí.

## **I. TEORETICKÁ ČÁST**

## 1 INTELIGENTNÍ BUDOVA

Mezi často skloňované termíny automatizace ve stavebnictví patří v poslední době termín Inteligentní budovy. Pojem inteligentní budova – stejně jako mnoho jiných podobných pojmů – pochází ze Spojených států. Ustálená a přijímaná definice tohoto pojmu dosud neexistuje, takže se vyskytuje mnoho forem této definice. [10]

Pro orientaci zde uvedu několik nejzajímavějších definic:

### 1.1.1 Definice č. 1

Inteligentní budova je taková budova, která je schopná se přizpůsobit změnám ve způsobech jejího užívání a změnách životního stylu jejích obyvatel a nepřestává jim sloužit a vytvářet příjemné a odpovídající prostředí. [10]

### 1.1.2 Definice č. 2

Inteligentní budova je budovou plně vybavenou automatizační a informační a komunikační technikou, která slouží jednak přímo svým obyvatelům, jednak pro vytváření příjemného prostředí pro ně. [10]

### 1.1.3 Definice č. 3

Inteligentní budova je budova, v níž jsou jednotlivé inteligentní prvky či systémy integrovány a řízeny prostřednictvím jediného řídicího systému. [10]

Jak je vidět z jednotlivých definic, inteligentní budova je velice široký a obecný pojem. O něco konkrétnější je pojem inteligentní dům. O inteligentním domě pojednává následující kapitola.

## 2 INTELIGENTNÍ DŮM

Inteligentní dům je obecně budova, ve které se pohybuje člověk. Samotná inteligence domu znamená, že je budova řízena systémem, jehož cílem je vytváření, udržování a správa podmínek pro pobyt v prostorách budovy (anebo procesů v budově probíhajících) takových, že reagují na měnící se vnější podmínky anebo vnitřní či vnější požadavky. To vše s cílem optimální spotřeby energií a minimalizace nákladů. Právě uvědomělá a smysluplná reakce na měnící se podmínky bývá považována za „inteligenci“. [10]

### 2.1 Typický inteligentní dům

Hlavním cílem inteligentního domu je usnadnit a zpříjemnit uživatelům bydlení. Alespoň bez základní elektroniky, jako jsou termostaty pro řízení topení, osvětlení a zabezpečovací systém, se dnes neobejde žádný dům. V mnoha domech k tomu přibude počítačová síť, řízení rolet a žaluzií, klimatizace, kamerový systém, v neposlední řadě domácí kino a ozvučení alespoň některých prostor. [5]

Dnes běžný stav v „normálních“ domech je, že tato technika nedokáže navzájem spolupracovat, každý systém má jiný způsob ovládání a téměř vždy vzhledově naprosto odlišné ovládací prvky. [5]

Inteligentní dům dokáže všechnu techniku propojit a integrovat mezi sebou a sjednotit její ovládání, a to jak z hlediska vzhledu vypínačů a displejů na zdi, tak především poskytnout jednotný způsob ovládání, přizpůsobený na míru pro konkrétní dům a jeho obyvatele. [5]

### 2.2 Inteligentní elektroinstalace

Jedním ze stěžejních cílů inteligentního domu je, aby byly veškeré spotřebiče a jiné prvky navzájem propojené. Smyslem tohoto propojení je možnost ovládat vše centrálně z jediného místa. Propojení může být realizováno mnoha způsoby. Druhy propojení se dělí na hvězdicové a sběrníkové.

Hvězda znamená, že kabely od všech vypínačů, světel, rolet apod. se sbíhají do jednoho jediného centrálního místa. Tuto topologii používá například anténní rozvod a strukturovaná kabeláž – počítačová a telefonní síť. [5]

Při napojení na sběrnici vede mezi všemi částmi elektroinstalace pouze jediný kabel, tj. všechny vypínače a světla jsou postupně zapojeny v řadě za sebou. V minulosti byl tento způsob používán i pro počítačovou síť a rozvod k anténním zásuvkám, dnes jej najdeme třeba u inteligentní elektroinstalace podle standardu KNX/EIB nebo u EIA-485, používaného často pro automatizační zařízení v průmyslu. [5]



Obrázek 1: Řadový akční člen KNX/EIB. Řídí osm nezávislých žaluziových pohonů s možností ručního ovládání. [5]

### 3 POUŽITÉ SOFTWAREVÉ TECHNOLOGIE

Při práci na modelu byla použita široká škála různých softwarových technologií. Od programovacích jazyků a knihoven, až po uživatelské programy určené například pro návrhy plošných spojů. Veškerý použitý software je zdarma, případně je použita zkušební omezená verze, která je k dispozici zdarma.

#### 3.1 Programovací jazyk C++

Celý model domu bude řízen z počítače typu PC pomocí řídicí aplikace. Pro napsání této aplikace byl zvolen programovací jazyk C++.

Tento jazyk vychází z jazyka C, přičemž hlavní změnou je přidání podpory objektově orientovaného programování. Jazyk C++ se stal jedním z nejdůležitějších programovacích jazyků devadesátých let. Objektově orientované vlastnosti dávají tomuto jazyku novou metodiku programování, která je navržena tak, aby si poradila se stupňující se obtížností moderních úkolů programování. [11]

##### 3.1.1 Objektově orientované programování

U klasického strukturovaného programování je poměrně obtížné programování velkých celků. Tuto obtížnost je možné obejít díky přístupu, který přináší objektově orientované programování (OOP). Zjednodušeně řečeno, OOP klade důraz na organizaci dat. Naproti tomu klasické strukturované programování klade důraz spíše na algoritmy. Myšlenka OOP spočívá v návrhu datových forem, které odpovídají základním rysům problému. [11]

Při psaní programů s využitím OOP nejprve navrhujeme třídy, které přesně popisují to, s čím program pracuje. [11]

#### 3.2 Knihovna wxWidgets

Jak již bylo zmíněno, model domu bude řízen pomocí aplikace z PC, připojeného k modelu přes USB kabel. Tato aplikace je určena pro širokou veřejnost, proto nepřipadá v úvahu ji

udělat jako konzolovou aplikaci. Bylo rozhodnuto udělat okenní aplikaci pro systém Windows. K programování okenních aplikací lze využít přímo API rozhraní Windows. To je ale poměrně složité a moc se nevyužívá. Lepší, efektivnější a jednodušší je použít nějakou z knihoven, která se zabývá prací s grafickým rozhraním a přináší jednoduché funkce pro vytváření okenních aplikací.

Knihovna wxWidgets je multiplatformní knihovna pro jazyk C++, která umožňuje programátorovi vytvářet okenní aplikace pro Windows, Mac OS X, Linux a další. Tato knihovna používá na dané platformě její nativní API, namísto emulace grafického uživatelského rozhraní. Díky tomu vypadají aplikace „přirozeněji“. Kromě C++ je možné knihovnu použít i v několika dalších rozšířených jazycích, jako jsou Python, C#, Haskell, Lua, Perl, Ruby, Smalltalk, Java, JavaScript.

Knihovna wxWidgets je přímo vestavěna do vývojového prostředí wxDevC++. To je výhodné, protože díky tomu lze v tomto prostředí začít snadno a rychle vyvíjet okenní aplikace. Prostředí také obsahuje generátor kódu pro tuto knihovnu, což znamená, že si uživatel v grafickém editoru snadno navrhne vzhled své okenní aplikace, a prostředí samo vygeneruje patřičný kód implementující zadaný vzhled pomocí knihovny wxWidgets.

### 3.3 Vývojové prostředí Arduino

Do modelu domu bude vestavěn mikropočítač, který bude komunikovat s řídicí aplikací v osobním počítači, a to přes USB propojení. Mikropočítač v dnešní době představuje samostatný integrovaný obvod, který obsahuje mikroprocesor a paměť. [4]

Díky široké škále různých vývojových prostředí a systémů je v dnešní době možné programovat mikropočítače téměř v jakémkoliv rozšířeném programovacím jazyce. Asi nejrozšířenější jsou jazyky C a C++. Těmto jazykům je velmi podobný jazyk Wiring, z něhož vychází vývojové programovací prostředí Arduino. [3]

Wiring je open-source programovací framework pro mikropočítače. Umožňuje vytvářet multiplatformní programy, sloužící pro řízení zařízení připojených k široké paletě mikropočítačů. [12]

Jazyk Arduino vychází z jazyka Wiring a je mu až na drobné detaily velmi podobný. Změny spočívají hlavně v přizpůsobení jazyka tak, aby byl snadno a rychle použitelný i pro úplné začátečníky.

V této práci je Arduino jazyk využit pro programování řídicího mikropočítače v modelu domu. Stejně jako v C++ je zde k dispozici OOP, což dělá z tohoto jazyka velmi silný nástroj na poli mikropočítačů. Dříve se k takovému programování používal hlavně Assembler, později i obyčejné C. S nástupem OOP do oblasti mikropočítačů se vše posouvá na novou úroveň. Je ovšem potřeba počítat i s tím, že čím „vyšší“ jazyk, tím více místa program zabere v paměti mikropočítače, a s touto pamětí je potřeba šetřit. Nicméně moderní mikropočítače mívají již dostatek paměti, takže si u běžných projektů programátor nemusí hlídat velikost programu.



Obrázek 2: Náhled na vývojové prostředí Arduino

### 3.4 EAGLE

V modelu domu je obsaženo velké množství elektroniky. Ta je umístěna na desce s plošným spojem, který byl kompletně navrhnout a vyroben svépomocí. K tomu bylo potřeba nějaký navrhovací software, kterých existuje mnoho. Já jsem se vzhledem ke svým předchozím zkušenostem rozhodl sáhnout po nástroji EAGLE.

Editor plošných spojů EAGLE je uživatelsky přívětivý a výkonný nástroj pro návrh desek plošných spojů (DPS, PCB). Název EAGLE je zkratka, pocházející z původního názvu - Easily Applicable Graphical Layout Editor. Program se skládá ze tří hlavních modulů: [13]

- Editor spojů
- Editor schémat
- Autorouter

#### 3.4.1 Editor schémat

Editor schémat je grafický editor, který slouží ke kreslení elektrických schémat. Je zde k dispozici celá řada knihoven, které obsahují nejrůznější elektrotechnické součástky. Při kreslení schématu se tyto součástky z knihoven vybírají a ve schématu se jim přidělují hodnoty a názvy. Každá součástka má uložené informace o jejích rozměrech a tvaru, aby bylo možné později ze schématu navrhnout desku plošného spoje.

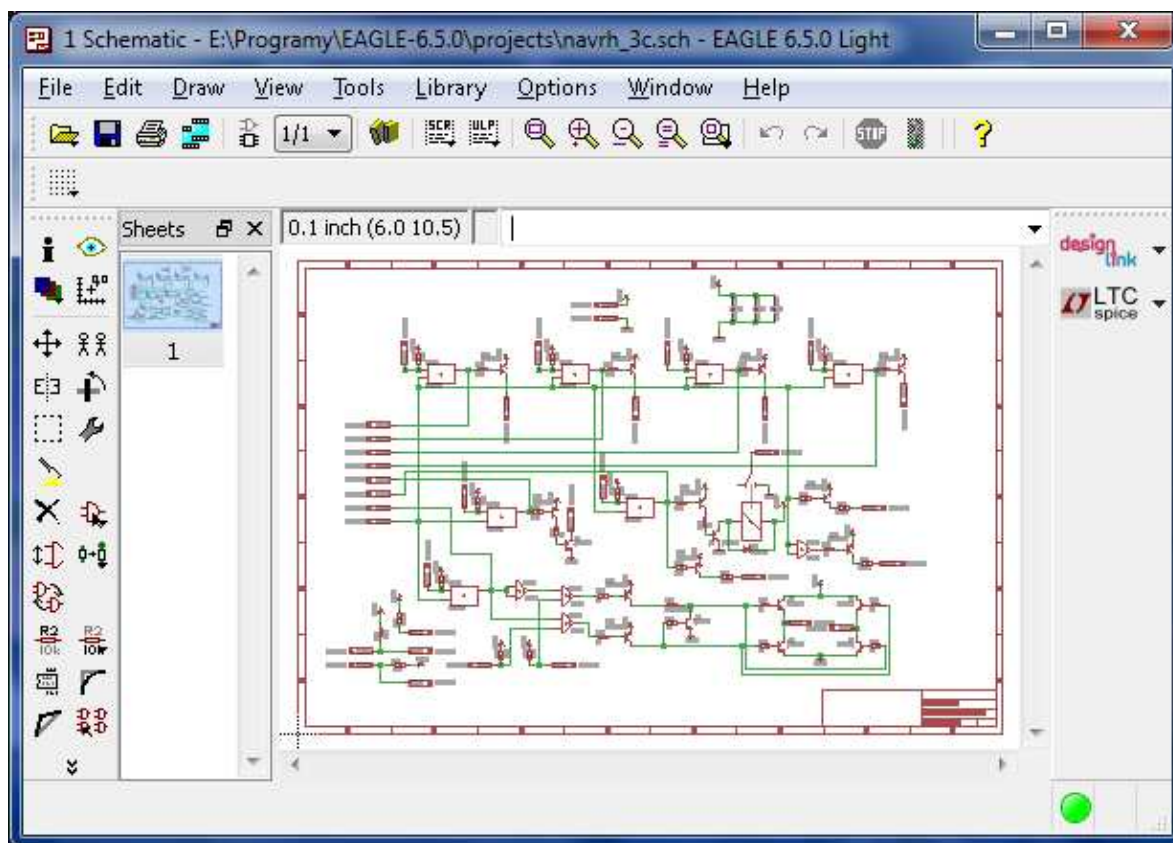
#### 3.4.2 Editor spojů

Po nakreslení schématu z něj lze vygenerovat desku spojů. K tomu slouží editor spojů. Ten má rovněž grafické rozhraní. V tomto editoru se rozmisťují součástky na desce a kreslí se spoje mezi nimi, tak jak budou na desce vypadat. Kreslené spoje se samozřejmě nesmí křížit, takže u složitějších schémat bývá návrh spojů touto manuální cestou velmi obtížný. S návrhem nám může pomoci autorouter.

### 3.4.3 Autorouter

Abychom nemuseli spoje v editoru spojů kreslit ručně, je v Eaglu k dispozici autorouter. Autorouter je mocný nástroj, který dovede potřebná spojení propojit automaticky. Nevýhodou je, že součástky musí vždy rozmístit uživatel. Právě správné rozmístění součástek má velký vliv na to, zda bude vůbec možné všechny spoje propojit. Autorouter tedy v žádném případě není zárukou úspěchu. Nedokáže totiž propojit desku, kde jsou uživatelem špatně rozmístěny součástky. Práce s autorouterem tedy většinou spočívá ve dvou krocích: ruční rozmístění prvků, spuštění autorouteru. Pokud autorouter nepropojí všechny spoje, vrátíme se opět na začátek a znova rozmístíme součástky.

Úspěšné propojení také velmi závisí na počtu spojových vrstev. V dnešní době jsou plošné spoje mnohvrstvé. U autorouteru lze počet vrstev nastavit. Ve zkušební verzi Eaglu lze použít maximálně dvě vrstvy.

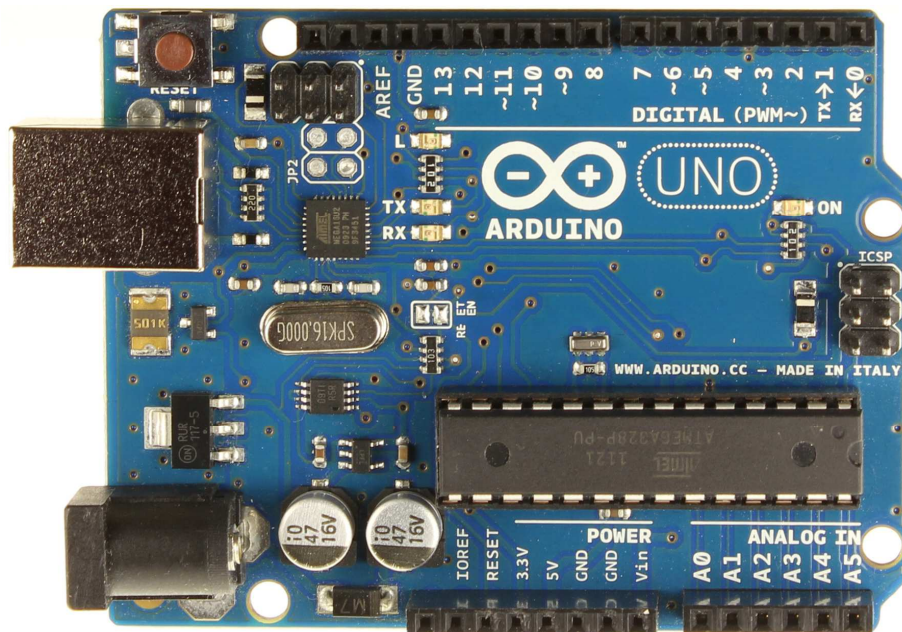


Obrázek 3: Náhled na editor schémat v systému EAGLE

## 4 VÝVOJOVÝ KIT ARDUINO

Mezi základní požadavky této práce patří, aby byl celý model řízen mikropočítačem. Jako nejvhodnější byl zvolen poměrně rozšířený a oblíbený vývojový kit Arduino Uno.

Arduino dokáže interagovat s prostředím pomocí připojených vstupních senzorů, a dokáže své okolí ovlivňovat spínáním různých akčních prvků, světel, motorů, a tak dále. Mikropočítač ve vývojovém kitu se programuje pomocí Arduino programovacího jazyka, založeného na jazyku C++. Software běžící v Arduino kitu může být plně samostatný, avšak může i za chodu komunikovat se softwarem běžícím na PC. Technická výrobní dokumentace k Arduino kitu je na internetu volně k dispozici, takže si může tento kit každý vyrobit sám, jednodušší a pravděpodobně i levnější je však koupit Arduino kit již hotový. Veškerá softwarová podpora a vývojové prostředí je zdarma na internetu. [9]



Obrázek 4: Vývojový kit Arduino Uno [9]

Kit Arduino Uno je nejzákladnější a nejrozšířenější verze. Obsahuje mikropočítač ATmega328. Má 14 digitálních vstupně/výstupních pinů, z nichž šest může být použito jako výstupy s pulzně-šířkovou modulací. Dále je zde šest analogových vstupů, které obsahují samostatné 10ti bitové AD převodníky. Mikropočítač je taktován 16 MHz

krystalovým oscilátorem. Kit má USB konektor pro připojení a komunikaci s PC, jack napájecí konektor pro napájení z externího zdroje a resetovací tlačítko. Provozní napětí kitu je 5 V. Doporučené vstupní napětí je v rozsahu 7 až 12 V. Limitní vstupní napětí je 6 až 20 V. Mikropočítač má FLASH paměť pro program o velikosti 32 KB, operační paměť SRAM o velikosti 2 KB a 1KB EEPROM paměti, sloužící jako miniaturní „pevný disk“.

[9]

Ke komunikaci s počítačem je kit sice vybaven USB konektorem, komunikace samotná však využívá protokol RS232. USB slouží jen jako hardwarový „mezistupeň“, a to z důvodu nedostupnosti sériových COM portů na moderních počítačích. Kit je vybaven druhým mikropočítačem ATmega16U2, který je naprogramován jako obousměrný převodník mezi USB a RS232 komunikací. V počítači je pak potřeba mít nainstalován USB ovladač kitu, který se z pohledu aplikací v PC tváří jako virtuální COM port. Veškeré programování – jak software do kitu, tak software na PC komunikující s kitem – je tedy možné psát stejně, jakoby veškerá komunikace probíhala klasicky přes COM port. [1]

Napájení kitu je možné přes USB konektor, přídatný napájecí konektor jack je zde pouze pro ty aplikace, které vyžadují napojení větší proudové zátěže na kit, nebo pro případ, kdy je kit využit jako samostatně fungující jednotka bez připojení k PC.

## **II. PRAKTICKÁ ČÁST**

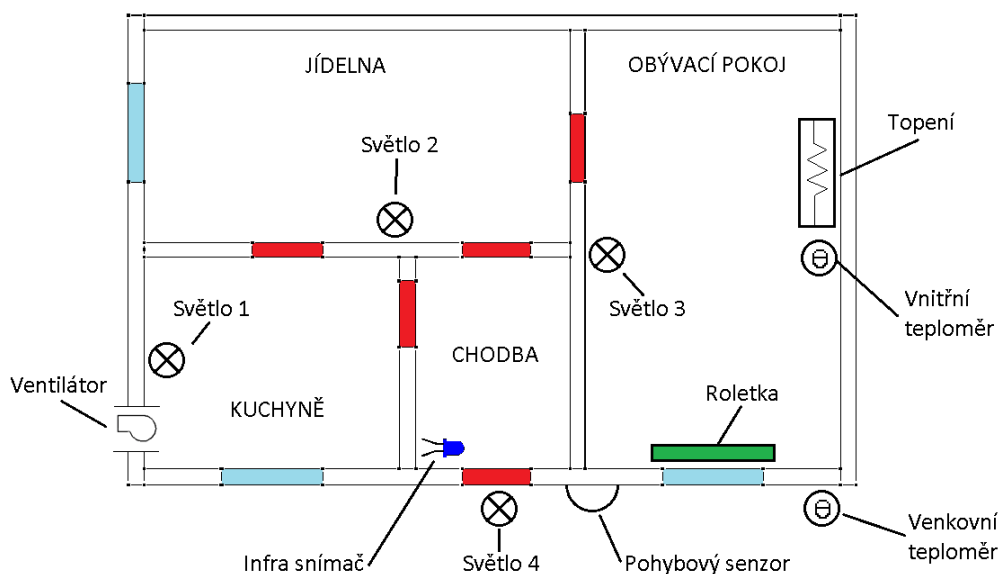
## 5 NÁVRH MODELU

Tato diplomová práce je vytvořena na základě reálného zadání od externího zadavatele. Jeho požadavek byl vytvořit funkční model inteligentního domu, určený jako exponát na výstavu „Experimentarium“ v Otrokovicích. Konkrétní návrh modelu byl vytvořen na základě konzultací s vedoucím diplomové práce.

### 5.1 Funkčnost modelu

Domek bude obsahovat tři místnosti a chodbu, jak je vidět na obrázku 5. Bude zde celkem 7 akčních prvků:

- světlo č. 1
- světlo č. 2
- světlo č. 3
- světlo č. 4 (venkovní světlo)
- ventilátor nad kuchyňskou linkou
- elektrické topení v obývacím pokoji
- elektricky ovladatelná roletka za oknem obývacího pokoje



Obrázek 5: Návrh a rozmístění místností a funkčních prvků

Domek bude dále obsahovat 4 snímače:

- infračervený snímač
- pohybový senzor
- venkovní teploměr
- vnitřní teploměr

Ovládání modelu bude mít dva režimy: automatické a manuální.

### **5.1.1 Manuální ovládání**

Všechny akční prvky bude možné zapínat a vypínat manuálně pomocí ovládacího panelu u modelu domku. Na tomto panelu bude také přepínač na volbu režimu ovládání.

### **5.1.2 Automatické ovládání**

V automatickém režimu se bude model chovat podle nastavených parametrů. Model bude připojen pomocí USB kabelu k osobnímu počítači, kde poběží řídicí aplikace. Odtud bude možné měnit tyto parametry. V aplikaci bude seznam akčních prvků, uživatel si zvolí, který z nich chce upravit. V hlavní části okna aplikace se zobrazí seznam měnitelných parametrů, včetně rozbalovacího menu s možností přiřazení libovolného senzoru. Půjde tedy například nastavit, aby se ventilátor sepnul při určité teplotě, nebo po otevření hlavních dveří. Každý akční prvek půjde sepnout buď hned po aktivaci zvoleného snímače, nebo s nastavitelným zpožděním, a to po určitou dobu. V případě volby teploměru jako snímače bude možné přesně definovat „spínající“ rozsah teplot.

## 5.2 Fyzický model domu

### 5.2.1 Použité materiály

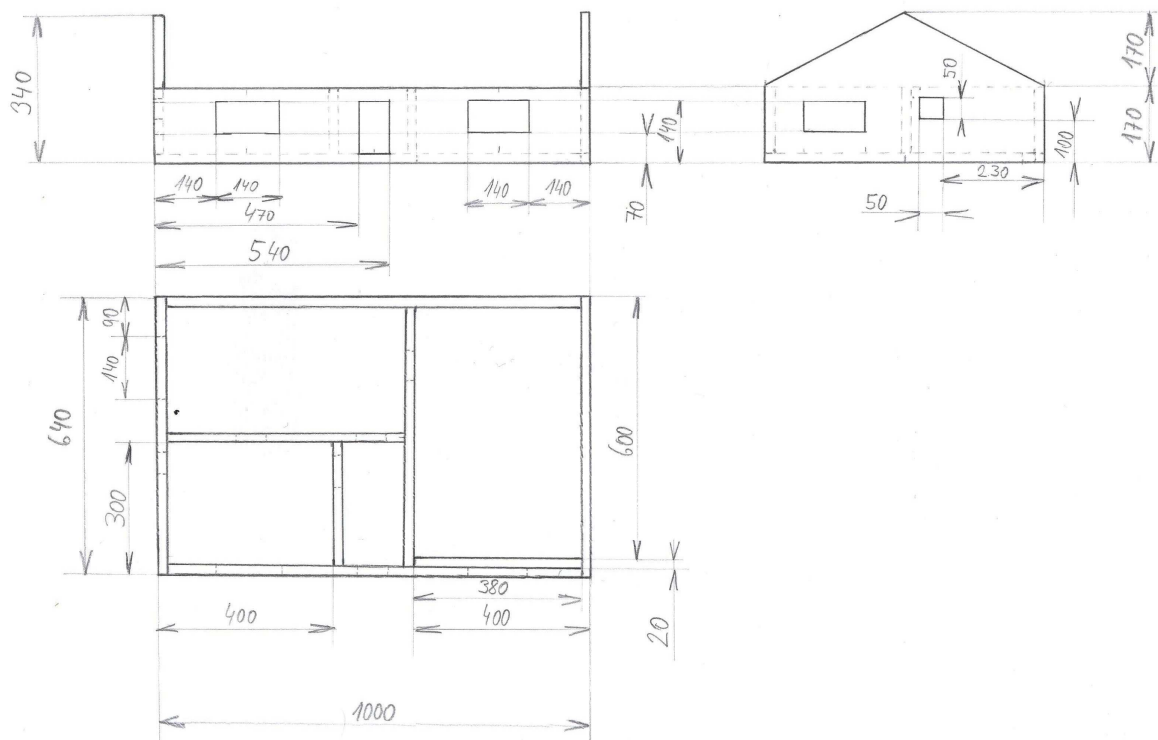
Můj původní návrh byl použit mnohovrstvou (multiplex) překližku. Tato mnohovrstvá překližka je na rozdíl od té obvyčejné vhodná i pro zatloukání hřebíků. Bohužel je velmi drahá a má nevzhledný povrch, který by bylo nutné dodatečně esteticky upravit.

Druhou možností bylo použití materiálu LDT, tedy laminované dřevotřískové desky. Je to deska vyrobená z dřevěných třísek, které jsou slepené fenolformaldehydovou pryskyřicí. Spodní i horní strany těchto desek jsou laminovány. Je na nich tedy nanesena vrstva melaminové pryskyřice a dekoračního papíru a vše je zalisováno. Má různé barvy nebo texturu dřeva. Díky současným grafickým možnostem bývají tyto imitace takřka nerozeznatelné od originálu. [6] Tento materiál je levnější než překližka, nicméně se nedá upravovat. Jako nepřekonatelný problém LDT desek se ukázalo, že je téměř nemožné do desky vyvrtat nebo vyřezat jakýkoliv otvor (díru, okno, dveře), aniž by nedošlo k nevratnému poškození laminovaného povrchu na odvrácené straně řezané desky.

Z výše popsaných důvodů bylo nakonec rozhodnuto použít materiál MDF. Desky z tohoto materiálu jsou vyrobeny bez použití lepidel. Slisovaná dřevovláknitá deska má velmi kompaktní strukturu a díky tomu se dobře tvarově zpracovává. Lze do ní snadno vrtat, frézovat, ale může být i hladká. [6] Přestože je uměle vyrobená, její povrch může vzhledem připomínat texturu dřeva. Tyto desky jsou navíc velmi levné, takže byly nakonec jasnou volbou.

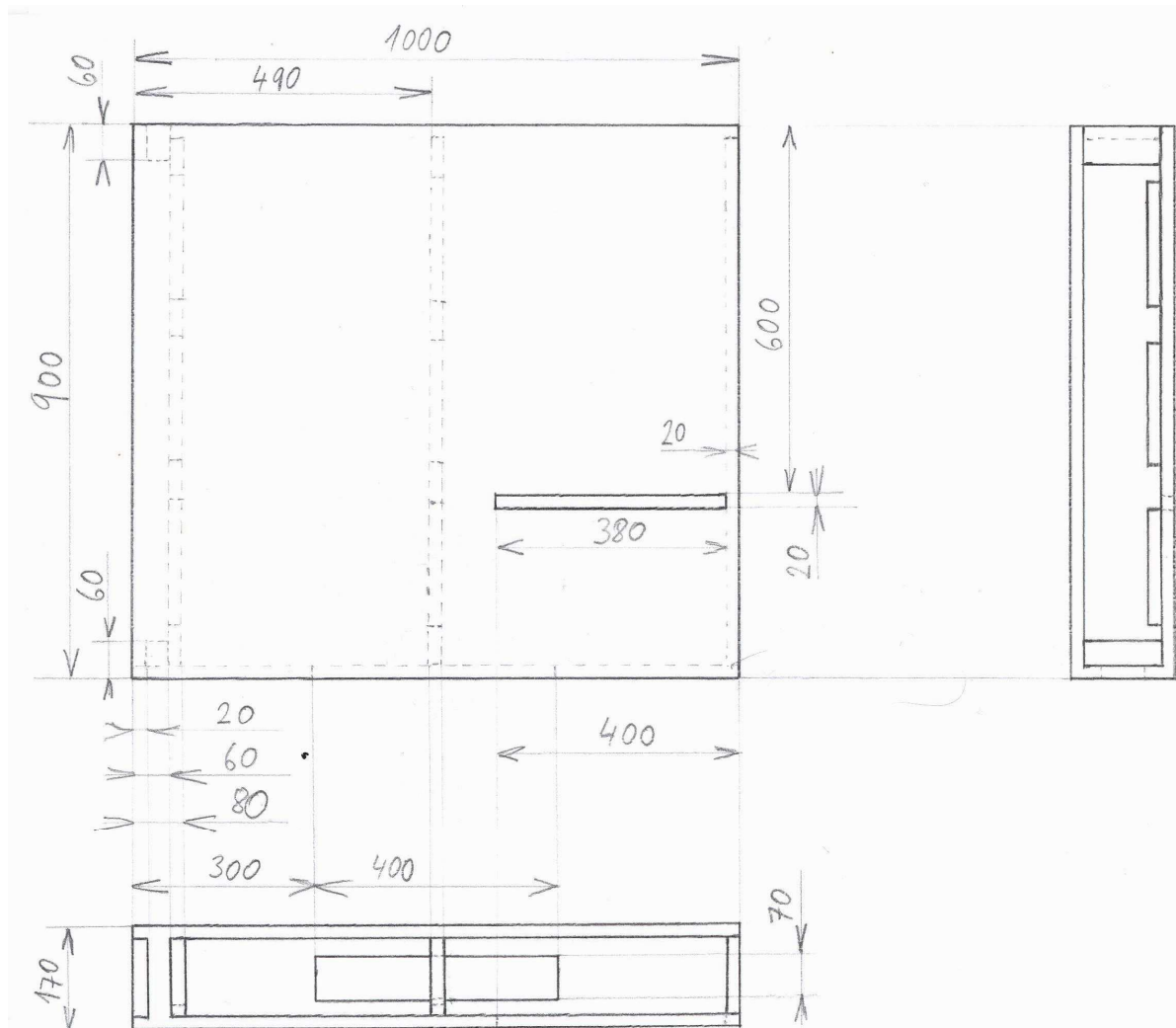
### 5.2.2 Technický návrh modelu

Model bylo rozhodnuto vyrobit poměrně veliký. Skládá se z podstavce a modelu budovy s přízemím a podkrovím. Celkové rozměry půdorysu včetně podstavce jsou 100 x 90 cm. Podkrovní prostor je prázdný a celá jedna polovina střechy je průhledná (chybí), aby bylo možné vidět skrz ni do jednotlivých místností. V podstavci se ukrývá většina elektroniky, rozvody drátů atd. V přízemí jsou celkem tři místnosti a jedna chodba. Z chodby vedou vchodové dveře.



Obrázek 6: Nárýs, půdorys a bokorys modelu domu

Na obrázku 6 je vidět nárýs, půdorys a bokorys modelu. Tato konstrukce byla navržena s ohledem na funkce navrhovaného modelu. Přízemí se skládá z kuchyně (vlevo dole), jídelny (vlevo nahoře), obývacího pokoje (vpravo) a vchodové chodby. Podkroví, jak už bylo zmíněno, je prázdné a slouží pouze k dodání estetického vzhledu domku.

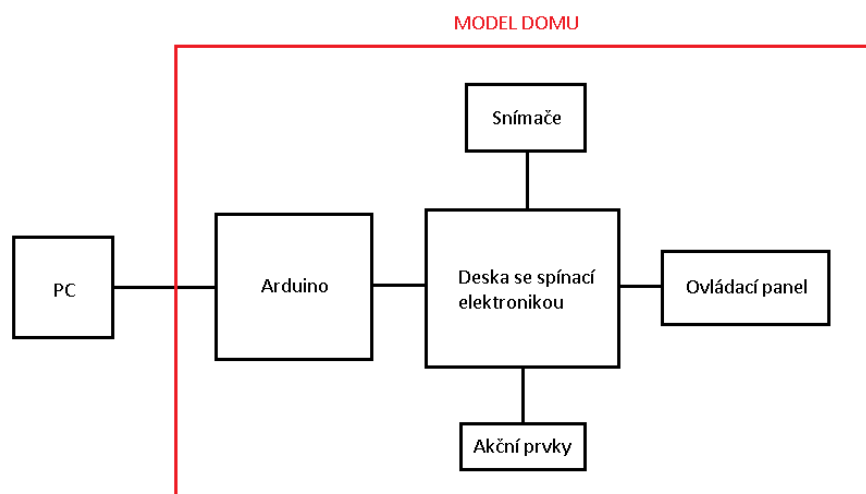


Obrázek 7: Nárys, půdorys a bokorys podstavce

Na obrázku 7 je vidět nárys, půdorys a bokorys podstavce. Model domu je umístěn na tomto podstavci. Hlavní účel podstavce je ukrytí kabeláže a řídicí elektroniky. Podstavec má vepředu obdélníkovou díru, která slouží pro instalaci předního panelu s ovládacími tlačítky. Na pravém boku podstavce je úzká vnitřní „místnost“ pro umístění desek plošných spojů. V pravé části podstavce je dlouhá úzká díra, která slouží pro umístění ovládacího mechanismu elektrické roletky.

## 6 ELEKTROINSTALACE V MODELU

Dlouho bylo promyšleno, jakou elektroniku do modelu zabudovat. Celkově to byl kompromis mezi realističností a jednoduchou proveditelností v modelovém měřítku. Hlavní požadavek byl, aby byla funkčnost modelu ovlivnitelná z nějakého vnějšího zařízení, například z PC. Proto bylo rozhodnuto použít v modelu programovatelný kit Arduino s mikropočítačem Atmel. Veškeré prvky v modelu jsou připojeny k tomuto kitu a jsou jím „centrálně“ řízeny.



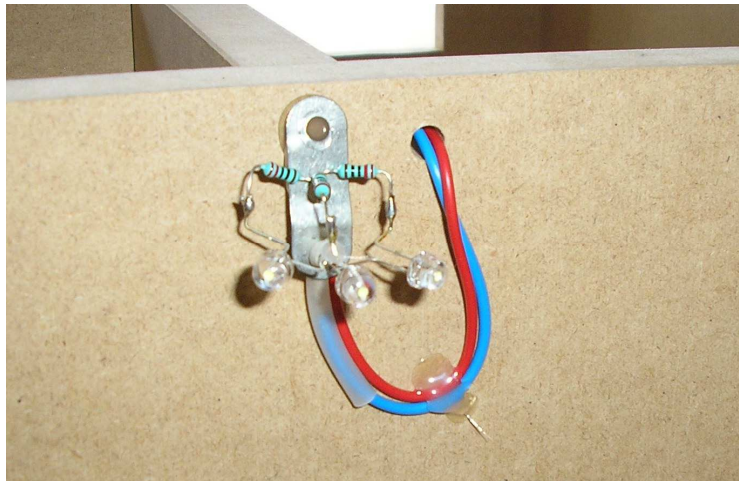
Obrázek 8: Blokové schéma elektroniky v modelu

### 6.1 Akční prvky

V modelu je celkem sedm akčních prvků. Čtyři světla, ventilátor, topení a elektrická roletka. Všechny tyto prvky jsou řízeny buď mikropočítačem, nebo manuálně pomocí přepínačů na předním ovládacím panelu. To v závislosti na tom, zda je nastaven manuální nebo automatický režim. V manuálním režimu lze jednotlivé prvky pouze zapínat a vypínat přepínači, kdežto v automatickém režimu se akční prvky řídí nastaveným programem. Program umožňuje spouštění prvků na základě aktivace různých snímačů.

### 6.1.1 Světla

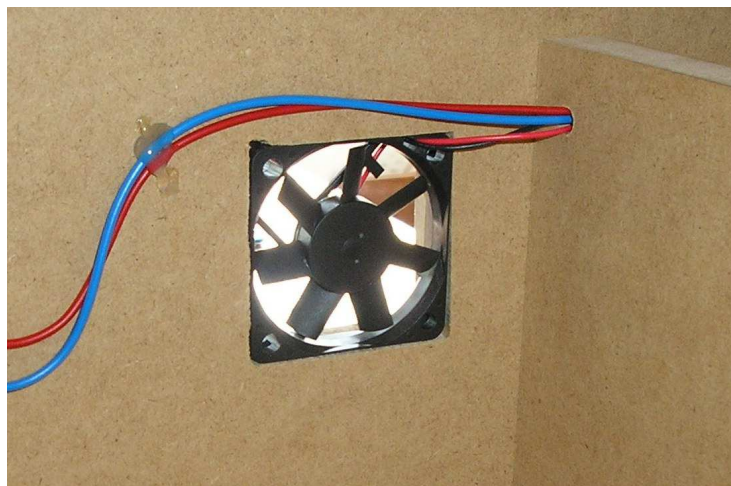
Každé ze čtyř světel je tvořeno třemi bílými LED diodami. Každá z diod má vlastní proudově omezovací rezistor, přičemž každá trojice je samostatně připojena na řídicí desku plošných spojů. Světla jsou po jednom umístěna v kuchyni, jídelně, obývacím pokoji a jedno slouží jako venkovní světlo nad hlavním vchodem.



Obrázek 9: Světlo v obývacím pokoji, tvořené třemi LED diodami

### 6.1.2 Ventilátor

Ventilátor má rozměry 40x40 milimetrů, vstupní napětí 12V. Je umístěn v kuchyni a plní účel odsávače par (digestoř) – fouká tedy směrem z kuchyně ven.



Obrázek 10: Ventilátor umístěný v kuchyni

### 6.1.3 Topení

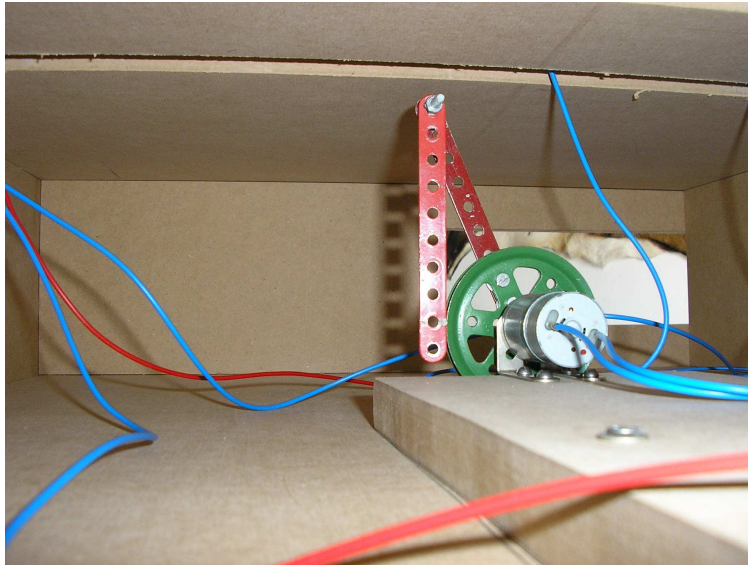
Topení je tvořeno dvěma topnými rezistory s vlastním chladičem. Každý z rezistorů má 10  $\Omega$  a vydrží 25 W. Oba rezistory jsou zapojeny do série a jsou napájeny napětím 12V. Topení je umístěno v obývacím pokoji a má signalizační červenou LED, aby měl uživatel „vizuální kontrolu“ nad stavem topení.



Obrázek 11: Výkonové rezistory použité jako topné těleso

### 6.1.4 Roletka

Roletka je kus látky vyztužený kostrou, celá je zasazena do vodících kolejnic. Pomocí kovového táhla připojeného na pákový mechanismus je spojena s motorkem. Motorek je vybaven převodovkou, takže jeho výstupní hřídelka má rychlost otáčení velmi malou (jedna otáčka za tři vteřiny), přičemž má dostatečnou sílu na uzdvihnutí roletky přes pákový mechanismus. Tento mechanismus i táhlo jsou vyrobeny ze stavebnice Merkur. Roletka pak má dva dojezdové snímače – obyčejné mikrospínače s malou kladkou pro snížení odporu. Na základě těchto snímačů elektronika vyhodnocuje, zda je roletka plně otevřená, nebo plně zavřená, respektive zda má již dojít k zastavení pohybu motorku. Toto řízení se děje čistě hardwarově za pomoci logických hradel. Zastavování tedy není závislé na programu, což snižuje riziko poškození motorku nebo roletky. Roletka je umístěna za oknem obývacího pokoje, přičemž celý mechanismus včetně motorku je umístěn v podstavci a propojen s roletkou skrz dlouhý, úzký montážní otvor v podlaze.



Obrázek 12: Motorek s pákovým mechanismem pro vytahování roletky – umístěno v podstavci

## 6.2 Snímače

V modelu jsou celkem čtyři snímače. Dva teplotní snímače, pohybový snímač a infračervený snímač. V manuálním režimu nemají tyto snímače žádnou funkci. V automatickém režimu mohou být využity jako vstupy pro spínání jednotlivých akčních prvků.

### 6.2.1 Teplotní snímače

Teplotní snímače jsou tvořeny integrovaným obvodem LM35. Tento obvod má 5 V napájení a jeden analogový napěťový výstup. Na tomto výstupu obvod poskytuje informaci o naměřené teplotě. Teplotní rozsah snímače je od  $-55\text{ }^{\circ}\text{C}$  do  $150\text{ }^{\circ}\text{C}$ , což je pro účely modelu domu více než dostatečné. Napěťový výstup snímače je po 10 mV na  $1\text{ }^{\circ}\text{C}$ , přičemž přesnost garantována výrobcem je  $0.5\text{ }^{\circ}\text{C}$ . Teplotní snímače jsou v modelu umístěny jeden v obývacím pokoji, druhý venku u okénka obývacího pokoje. Vnitřní snímač je umístěn přímo nad topením, a to z důvodu snadné a rychlé detekce změny teploty po zapnutí topení.



aplikován filtr záření, který propouští infračervené záření o vlnových délkách v rozsahu 8 až 14  $\mu\text{m}$ . Lidské tělo emituje do prostoru záření o vlnové délce 9,4  $\mu\text{m}$ . [7]

PIR snímač je ve venkovním prostoru modelu, a to mezi vchodovými dveřmi a oknem obývacího pokoje.

### 6.2.3 Infračervený snímač

Infračervený snímač použitý v modelu domu je v podstatě obyčejná infra závora. Jedná se o soustavu infra LED diody emitující infračervené světlo ( 940 nm ) a infra fototranzistoru, spínajícím při ozáření infračerveným světlem ( citlivost na 940 nm ). Přičemž tyto prvky jsou namířeny proti sobě. Takto vzniklá infra závora tedy detekuje přítomnost nějaké neprůsvitné překážky nacházející se mezi LED diodou a fototranzistorem. V modelu domu je tento infračervený snímač použit k detekci otevření hlavních dveří.

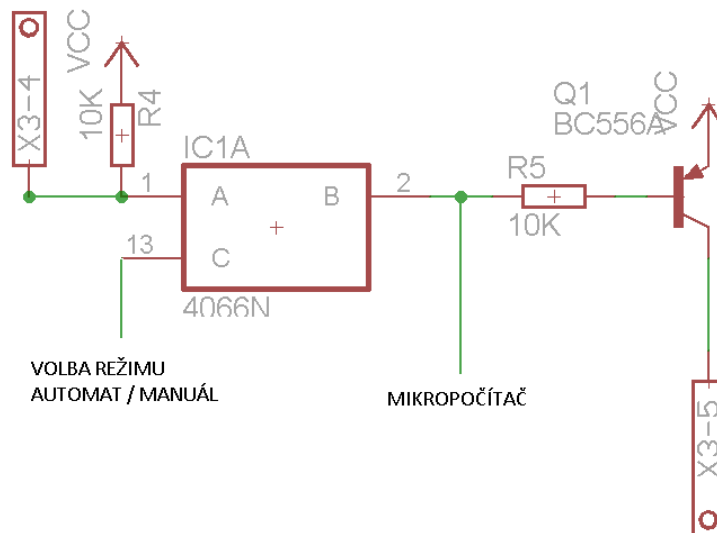
## 6.3 Plošný spoj se spínací elektronikou

Arduino kit poskytuje logické vstupy/výstupy s 5 V logikou a velmi nízkými proudy. Maximální proud procházející jedním datovým pinem je 40 mA. Toto je přitom pouze absolutní maximum jednoho pinu, nelze například zapojit kit tak, že poteče těchto 40 mA všemi piny! Z těchto důvodů nelze většinu elektronických prvků připojit přímo ke kitu. Je zde potřeba jakýsi mezistupeň – dalo by se říci zesilovač. Protože této pomocné elektroniky je v modelu potřeba hodně, rozhodl jsem se vše realizovat na vlastní plošný spoj.

### 6.3.1 Spínání světel

Spínání jednotlivých světel je řešeno pomocí PNP tranzistoru, který je využit jako spínač, a spíná 5 V napájení na LED diody. Všechny tři LED diody tvořící jedno světlo spotřebovávají dohromady přibližně 20 mA. PNP tranzistor jsem zvolil BC556A, jehož maximální proud kolektorem je 100 mA, což je dostatečné. Proud do báze je omezen bázovým rezistorem o velikosti 10 k $\Omega$ , bázový proud tedy nemůže překročit 0,5 mA, což je zároveň maximální proud, který bude tímto spínáním odebírán z portu mikropočítače. Při

spínání PNP tranzistorem teče bázev proud směrem z tranzistoru ven, světla se tedy spínají logickou nulou.



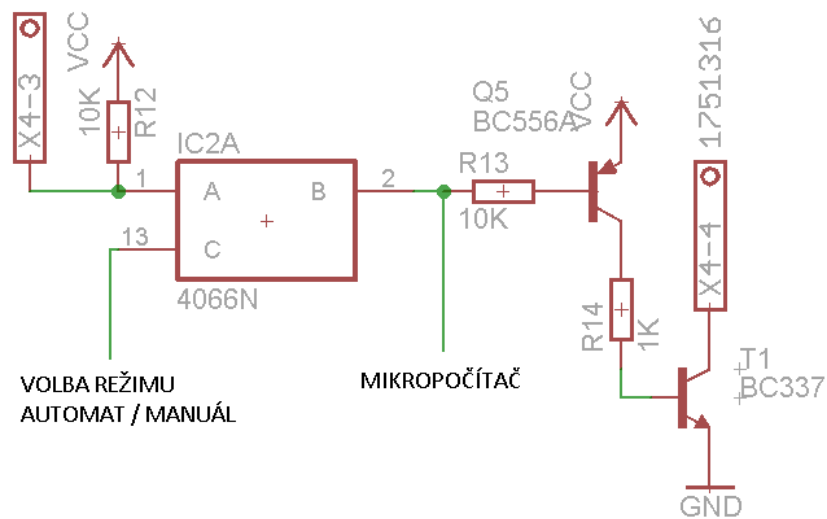
Obrázek 15: Schéma zapojení spínací elektroniky pro spínání světla

Na obrázku 15 je vidět, jak je zapojeno světlo (svorkovnice X3-5) k mikropočítači. Integrovaný obvod 4066N slouží k odpojení vodivé cesty mezi tlačítkem (svorkovnice X3-4) a mikropočítačem, v závislosti na tom, zda je zvolen automatický nebo manuální režim. Samotné elektronické spínání probíhá v pravé části obrázku, proud teče přes rezistor R5 do portu mikropočítače (nebo přes tlačítko na zem v manuálním režimu). 5 V napájecí větev je tak připojena na LED diody. V případě manuálního režimu a nestisknutého tlačítka drží rezistor R4 logickou jedničku, tedy uzavřený stav.

### 6.3.2 Spínání ventilátoru

Spínání ventilátoru je o něco komplikovanější. Ventilátor je totiž na 12 V. Bylo potřeba použít NPN tranzistor, a ventilátor připojit mezi kolektor a 12 V větev. Spotřeba ventilátoru je 80 mA, NPN tranzistor jsem zvolil BC337 s maximálním proudem kolektoru 800 mA. Při spínání NPN teče bázev proud směrem do tranzistoru. Proud do báze NPN tranzistoru je sepnut dalším tranzistorem, tentokrát PNP, a to podobně jako v případě spínání proudu do světla. Vše je vidět na obrázku 16, kde je opět stejný způsob řešení

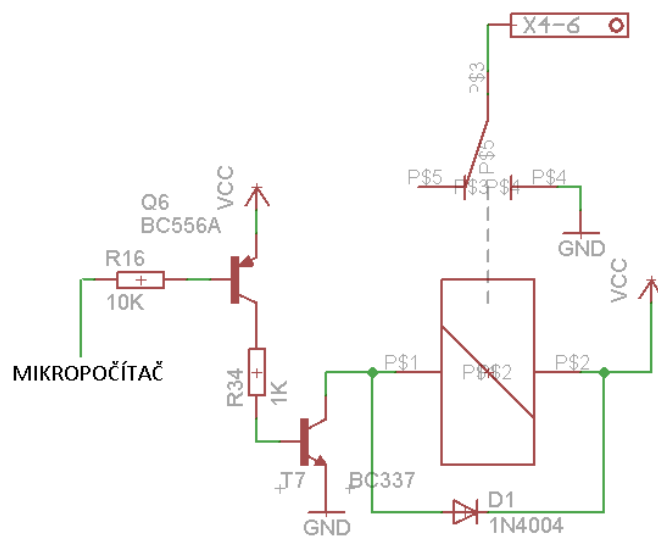
volby mezi automatickým a manuálním režimem. Tlačítko je na svorkovnici X4-3, zem ventilátoru je na svorkovnici X4-4.



Obrázek 16: Schéma zapojení spínání ventilátoru

### 6.3.3 Spínání topení

Dalším prvkem, který je potřeba spínat přes přídavnou elektroniku, je topení. Topení je tvořeno výkonovými rezistory, a je napojeno na 12 V. Je konstruováno tak (rezistory v sérii), aby jím protékalo 0,6 A. Jeho elektrický příkon je tedy 7,2 W. Vzhledem k velikému proudu jsem zde zvolil spínání přes relé. Zvolil jsem relé LT 05G, které má cívku na 5 V, ideálně se tedy hodí do navrhované elektroniky.

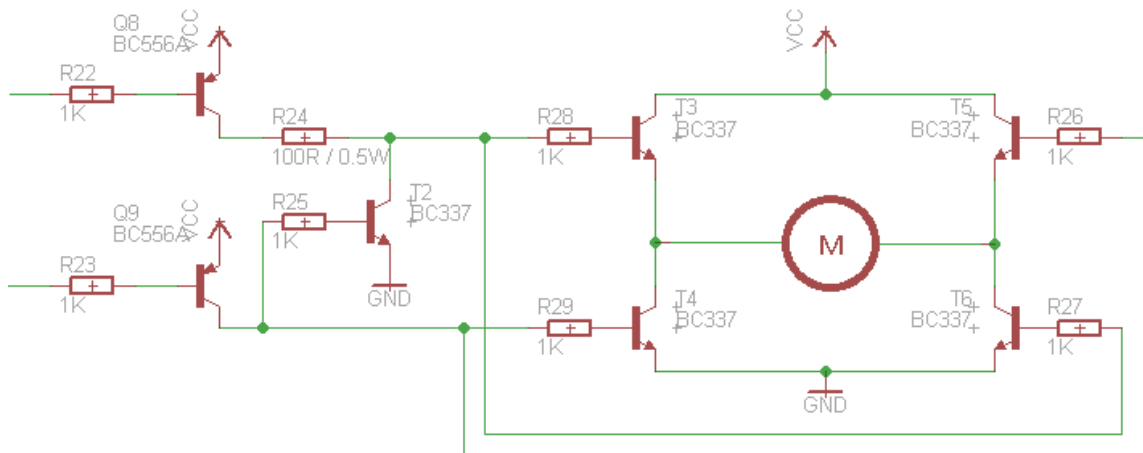


Obrázek 17: Schéma zapojení spínacího obvodu pro relé

Obrázek 17 znázorňuje, jak je elektronicky spínáno relé, které zapíná topení. Relé je spínáno poněkud nestandardně přes tranzistory NPN a PNP na zem. Mým požadavkem je co nejméně zatížit porty mikropočítače. Z tohoto důvodu se u spínání všech prvků napájí báze spínajícího tranzistoru přes velký odpor 10 k $\Omega$ . Relé však požaduje poměrně velký proud, a mnou provedené testy různých zapojení prokázaly, že PNP tranzistor BC556A s bázovým odporem 10 k $\Omega$  je pro účely cívky relé nedostatečně otevřen. Proto jsem se rozhodl tímto tranzistorem nespínat přímo cívku relé, ale pouze bázi dalšího tranzistoru přes 1 k $\Omega$  odpor. U tohoto zapojení je NPN tranzistor dostatečně otevřen a cívka se zapíná spolehlivě. Paralelně k cívce relé je připojena dioda v závěrném směru. Je to z toho důvodu, že se na cívce v důsledku přechodových jevů objeví po odpojení napájení opačné napětí o stejné velikosti jako napájecí napětí. Toto napětí by za normálních okolností mělo nepříznivý vliv na spínající tranzistor, v nejhorších případech by mohlo dokonce dojít k jeho destrukci. K zabránění vzniku této situace se paralelně k cívce vždy umísťuje tato ochranná dioda, která veškeré vzniklé opačně orientované napětí okamžitě vyzkratuje. Zem topení je připojena na svorkovnici X4-6, kladný pól topení je připojen na 12 V napájecí větev. Po sepnutí kontaktů relé se k topení připojí elektrická zem a dojde k jeho zapnutí.

#### 6.3.4 Spínání roletky

Dalším spínaným prvkem je elektrická roletka. Zde je situace nejsložitější. Roletka je poháněna motorkem na 5 V. Nejde zde však pouze o jeho spínání, je potřeba také volit, kterým směrem se má otáčet, a to změnou polarity jeho napájení. K tomuto účelu je zde H můstek, vytvořený ze čtyř tranzistorů, přes které teče proud do motorku. Je-li motorek vypnutý, jsou všechny čtyři tranzistory zavřené. Při zapnutém motorku jsou otevřeny vždy jen dva tranzistory, a to tranzistory ležící naproti sobě na diagonále. Podle toho která dvojice tranzistorů je otevřená, točí se motorek buď na jednu nebo na druhou stranu. Tranzistory v jednotlivých dvojicích se otevírají současně přes 1 k $\Omega$  rezistory, které jsou napájené přes dva PNP tranzistory, jak je vidět na obrázku 18 vlevo.



Obrázek 18: Schéma zapojení obvodu pro spínání motorku elektrické roletky

Tranzistory v H můstku, které spínají motorek, se nesmí nikdy otevřít všechny čtyři najednou, v takovém případě by došlo k přímému zkratu napájecího napětí. Pro ošetření této situace, respektive zamezení vzniku této situace, je v obvodu tranzistor T2 a rezistory R24 a R25. V případě, že by z vnějšku došlo k pokusu o otevření obou dvojic tranzistorů v H můstku, uzavře tento podobvod horní cestu přes rezistor R24 na zem. K H můstku se v takovém případě dostane „požadavek“ k otevření pouze jedné dvojice.

Drobným problémem tohoto řešení je, že vyžaduje napojení na dva piny, jsou zde totiž ne dva, ale tři stavy – vypnutý motorek, zapnutý motorek jedním směrem, zapnutý motorek druhým směrem (a čtvrtý zakázaný stav – zapnutý motorek oběma směry). Požadavek na jednoduchost softwaru vyžadoval připojení roletky pouze na jeden pin, tedy dva stavy – roletka nahoře, roletka dole. Proto je k ovládání roletky připojeno ještě další zapojení a dva dojezdové snímače. Jeden snímač je sepnut, když je roletka nahoře. Druhý snímač, když je dole. Roletka je tedy řízena ne mikropočítačem, ale kombinačním logickým obvodem, který má na svém vstupu informaci z obou dojezdových snímačů, a jednobitovou informaci o požadovaném stavu roletky.

### 6.3.5 Návrh kombinačního obvodu

Kombinační obvod bude mít tři vstupy:

A – požadovaný stav (1 = nahore, 0 = dole)

S1 – horní dojezdový snímač

S2 – spodní dojezdový snímač

Výstupy obvodu budou dva – viz výše (ovládání motorku)

M1, M2

Významy M1 a M2 výstupů jsou následující

M1 = 0, M2 = 1      =>      roletka jede nahoru

M1 = 1, M2 = 0      =>      roletka jede dolů

M1 = 0, M2 = 0      =>      roletka stojí

M1 = 1, M2 = 1      =>      zakázaný stav, nesmí nikdy nastat

Kombinační obvod bude mít celkem tři vstupy. Počet řádků pravdivostní tabulky vypočítáme jako  $2^n$ , kde n je počet vstupů. Tedy  $2^3 = 8$ , takže tabulka bude mít osm řádků.

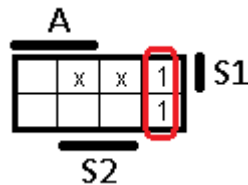
A	S1	S2	M1	M2
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	x	x
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	x	x

Tabulka 1: Pravdivostní tabulka kombinačního logického obvodu pro řízení roletky

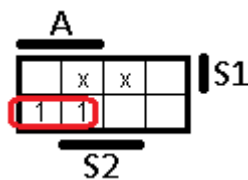
Jak vidíme v tabulce 1, pravdivostní tabulka je vyplněná podle výše popsané logiky řízení roletky, a to včetně zakázaných stavů.

Pro každý výstup je sestavena jedna Karnaughova mapa.

M1:



M2:



Pro každý ze vstupů je na základě příslušné Karnaughovy mapy sestavena jedna logická rovnice.

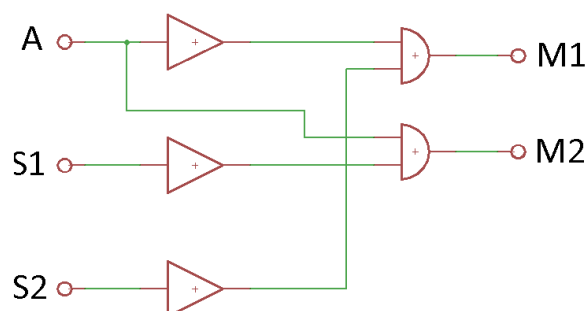
Rovnice pro M1:

$$M1 = \text{non}(A) \cdot \text{non}(S2)$$

Rovnice pro M2:

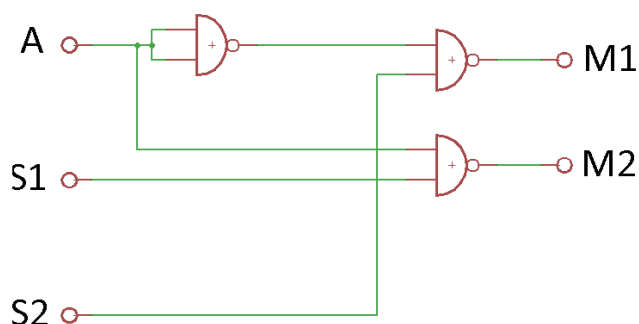
$$M2 = A \cdot \text{non}(S1)$$

Na základě logických rovnic byl sestaven tento kombinační obvod:



Obrázek 19: Kombinační obvod podle návrhu

Kombinační obvod vzniklý vlastním návrhem lze optimalizovat. Původní návrh očekává signalizaci dojezdových snímačů logickou jedničkou. Snímače je však jednodušší přizpůsobit hardwarově tak, aby se sepnutím dostaly do stavu logické nuly. Touto změnou se v kombinačním obvodu odstraní dva invertory u S1 a S2. U motorku se ve skutečnosti spínají dvojice tranzistorů logickou nulou, s čímž původní návrh pravdivostní tabulky nepočítal, protože byl sestaven ještě před návrhem schématu řídicího obvodu. Díky tomu je nutné zaměnit obě hradla AND za hradla NAND. Pro optimalizaci na počet pouzder je výhodné jediný zbývající invertor nahradit hradlem NAND se spojenými vstupy. Po těchto úpravách vznikne nový optimálnější kombinační obvod, jak je vidět na obrázku 20.



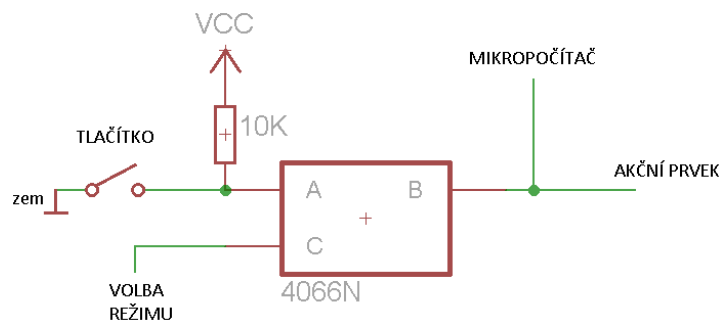
Obrázek 20: Optimalizovaný kombinační obvod

K mikropočítači bude tedy celá elektronika roletky připojena pouze jedním pinem, kterým bude program jednoduše určovat, zda má být roletka nahoře nebo dole.

### 6.3.6 Zapouzdření spínací elektroniky

Veškerá výše popsaná spínací elektronika je „zapouzdřena“ do jediné desky plošného spoje. Všechny akční prvky jsou spínány buď z mikropočítače, nebo manuálně tlačítkem. V tomto plošném spoji je zabudována i elektronika obsluhující přepínání mezi těmito režimy spínání. Přepínání je (jak bylo zmíněno dříve) vyřešeno následujícím způsobem.

Každý akční prvek je spínán logickou úrovní přes nějakou formu zesilovače. Tato vstupní logická úroveň je připojena přímo k mikropočítači, a zároveň přes integrovaný elektronický spínač 4066N k tlačítku. Obvod 4066N má podobný smysl jako relé.



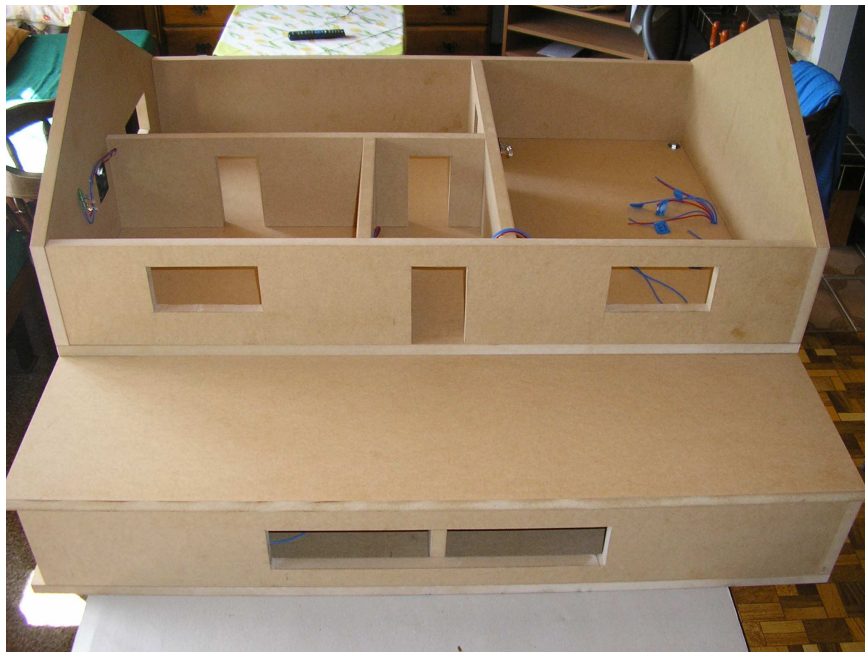
Obrázek 21: Schéma zapojení obvodu pro volbu spínacího režimu

Toto elektronické „relé“ je spínáno logickou jedničkou na vstupu C. A a B jsou spínané kontakty, mezi kterými je v rozepnutém stavu vysoká impedance. Řídící vstup C má velmi vysokou vstupní impedanci, cca bilion  $\Omega$  ( $10^{12} \Omega$ ), díky čemuž prakticky nezatěžuje řídicí logiku – mikropočítač. Spínané kontakty propojují tlačítko s akčním prvkem. V automatickém režimu je napojen pouze mikropočítač přímo na akční prvek (přesněji řečeno na spínací obvod akčního prvku – není na obrázku). Řízení prvku má tedy plně pod kontrolou mikropočítač. V manuálním režimu jsou sepnuty kontakty A, B. K akčnímu prvku je tedy v tomto stavu připojen zároveň mikropočítač i tlačítko. Toto je zdánlivě nebezpečný stav, avšak řídicí software v mikropočítači s tím počítá, a vždy v manuálním režimu přepne svůj pin do režimu INPUT, tedy vstupní pin. Mikropočítač i akční prvek fungují v takovém případě jako vstupy, tudíž je po elektrické stránce vše v pořádku. Tento vstup do mikropočítače je v tomto režimu softwarem ignorován. Tlačítko je zapojeno s rezistorem tak, aby v sepnutém stavu dávalo logickou nulu, v rozepnutém stavu zde rezistor drží logickou jedničku.

## 7 VÝROBA

### 7.1 Výroba modelu domu

Na základě navržených výkresů a zvoleného materiálu jsem si nechal nařezat, opracovat a nakolíkovat jednotlivé díly modelu ve stolařské dílně. Výroba těchto dílů trvala přibližně týden. Hotové díly pak byly svépomocí sestaveny a slepeny lepidlem SOUDAL na lepení dřeva. Model domu a podstavce jsou vyrobeny zvlášť, a po sestavení a slepení je domek k podstavci přišroubován čtyřmi šrouby M8 o délce 50 mm.



Obrázek 22: Model domu po sestavení a slepení hotových dílů

### 7.2 Vyhotovení desky s plošným spojem

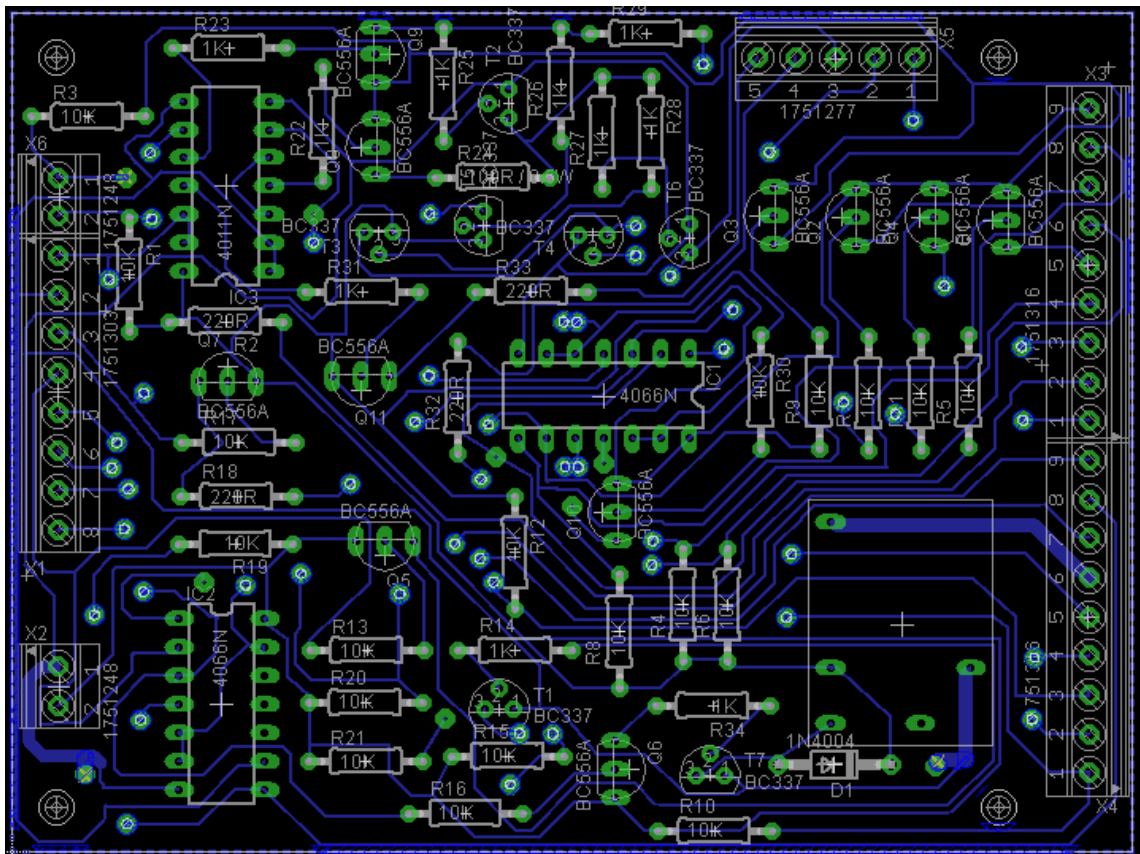
Veškerá doposud popisovaná spínací elektronika je realizována na jedné desce plošného spoje. V příloze P II je kompletní schéma této desky.

Ve schématu desky nejsou vidět žádné akční prvky, jsou zde místo nich zakresleny svorkovnice, které na desce budou a přes které bude deska s prvky propojena. Stejně tak je v levé části schématu vidět sada svorek, které budou propojeny s mikropočítačem (X1-1 až

X1-8, X6-1 a X6-2). Celé schéma je vyhotoveno v programu Eagle 6.5 Light, což je zkušební verze zdarma.

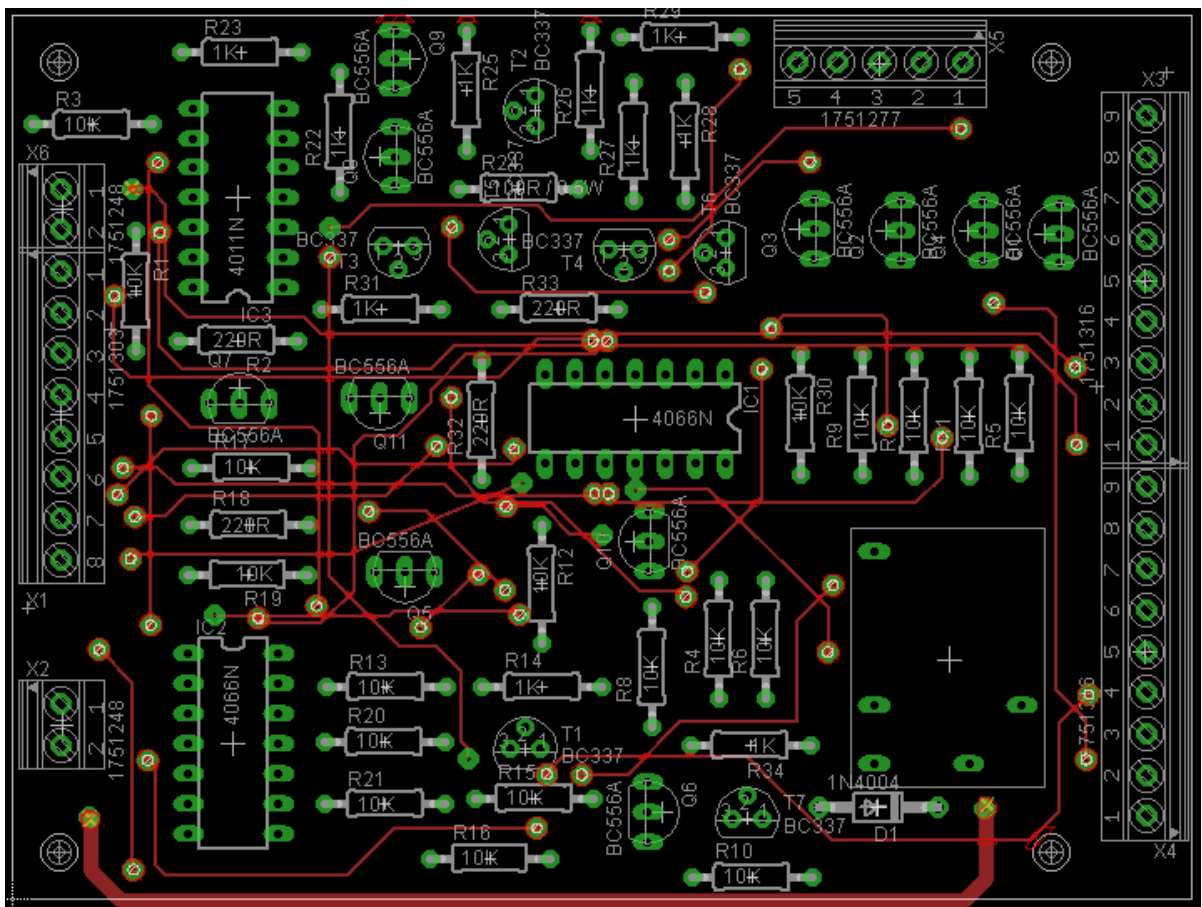
### 7.2.1 Návrh desky

Program Eagle umožňuje na základě nakresleného schématu navrhnout masku pro tištěný spoj. Návrh byl poměrně obtížný, protože Light verze Eaglu umožňuje velikost desky pouze 10x7,5 centimetrů. Vzhledem ke svým zkušenostem s výrobou plošných spojů jsem se rozhodl vyrobit pouze jednostranný plošný spoj. Vměstnat tolik součástek na tak malou plochu a propojit vše pouze jednou spojovou vrstvou je velmi obtížný úkol. Na obrázku 23 je vidět výsledný návrh strany spojů + zobrazení součástek.



Obrázek 23: Navržená deska plošného spoje – pohled ze strany součástek

Za zmínku stojí záměrně zesílené cesty určené pro proud procházející přes relé a napájecí topení. Tyto spoje bylo potřeba zesílit, protože zde bude protékat 0,6 A. Vzhledem k hustotě spojů na jediné spojové vrstvě se mi nepodařilo propojit 100 % všech spojení, ale pouze 80 %. Zbýlých 20 % spojů bylo nutno dořešit pomocí drátových propojek, umístěných na horní straně desky (strana se součástkami). Navržené propojky jsou vidět na obrázku 24. Na tomto obrázku opět stojí za zmínku zesílená drátová propojka dole, určená pro proud do topení.

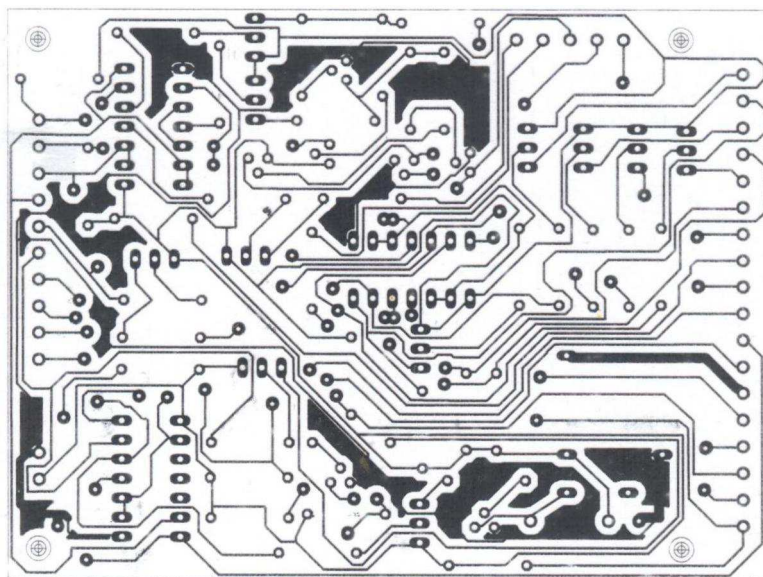


Obrázek 24: Zobrazení drátových propojek

Všechny tyto propojky bylo nutné dodatečně přidat do již hotové a osazené desky.

### 7.2.2 Výroba desky

Deska byla vyrobena fotocestou za použití pozitivní vývojky. To znamená, že je potřeba tištěný spoj vytisknout jako pozitiv na průhlednou fólii. Byla zvolena fólie určená pro tisk na laserových tiskárnách. Taková fólie totiž vydrží vysoké teploty, které na ni při tisku působí v zapékací fázi. Přímo z návrhového programu Eagle byla na tuto fólii vytisknuta předloha plošného spoje. V nastavení tisku byl zvolen režim SOLID a BLACK, aby se spoje vytiskly plnou černou barvou.



Obrázek 25: Předloha (maska) pro výrobu plošného spoje vytištěná na průhledné fólii

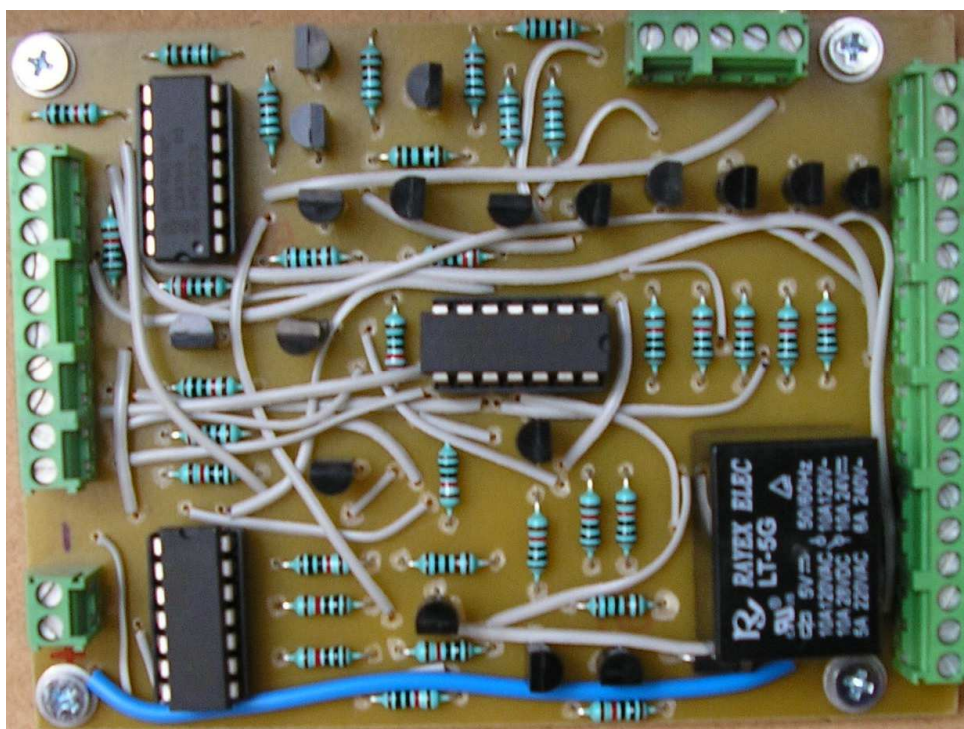
Bohužel, vzhledem k tiskovým vlastnostem laserových tiskáren v kombinaci s hladkým povrchem fólie nejsou černé plochy stoprocentně plné a neprůsvitné. Proto bylo potřeba masku ještě upravit. Pro slití toneru byla maska umístěna do uzavřené kartonové krabice, nad misku s acetonem. Zhruba po deseti minutách výpary z acetonu způsobily částečné slití toneru a vylepšení kvality masky. Poslední drobné nedostatky byly opraveny obyčejným lihovým fixem.

Hotová maska byla přiložena k cuprexitové desce s naneseným fotolakem. Následně byla deska s maskou osvětlena na 10 minut silným světlem z výbojky (horské sluníčko). Osvětlená deska byla vyvolána ve vývojce pro pozitivní fotoemulzi (roztok hydroxidu sodného). Celé osvětlování a vyvolávání probíhalo v temné místnosti s tlumeným

červeným světlem. Vyvolaná, opláchnutá a usušená deska byla vyleptána v roztoku chloridu železitého. Po vyleptání a umytí v ředidle byla deska hotová.

### 7.2.3 Osazení desky součástkami

Před osazením bylo nutné do desky navrtat všechny otvory pro pájení součástek. Všechny díry byly nejprve vyvrtány ostrým vrtákem o průměru 0,8 mm. Větší díry určené pro svorkovnice byly následně převrtány vrtákem o průměru 1 mm. Čtyři montážní otvory na krajích desky byly převrtány na rozměr 5 mm. Do předvrtané desky byly nejprve připájeny všechny součástky a následně všechny drátové propojky.



Obrázek 26: Hotová a osazená deska

### 7.2.4 Testování desky

Před konečnou montáží desky do modelu domu bylo potřeba desku důkladně otestovat. Pro tyto účely byla deska několika dráty připojena k nepájivému poli. K tomuto poli pak byly postupně připojovány různé akční prvky i snímače, a pomocí voltmetru, drátů a pull-up

rezistorů bylo uskutečněno simulované ovládání z mikropočítače. Po prvním testování bylo odhaleno velké množství chyb, převážně se jednalo o přeleptané spoje. Tyto chyby byly opraveny pomocí speciálních přídavných drátových propojek ze spodní desky (strana spojů). Další odhalenou chybou bylo špatně zapojené relé, které bylo připojeno na rozpínací kontakty místo spínacích. Relé bylo potřeba odpájet z desky a provést drobnou úpravu desky, uštípnout jeden pin relé a pak ho připájet zpátky.

### 7.3 Napájení

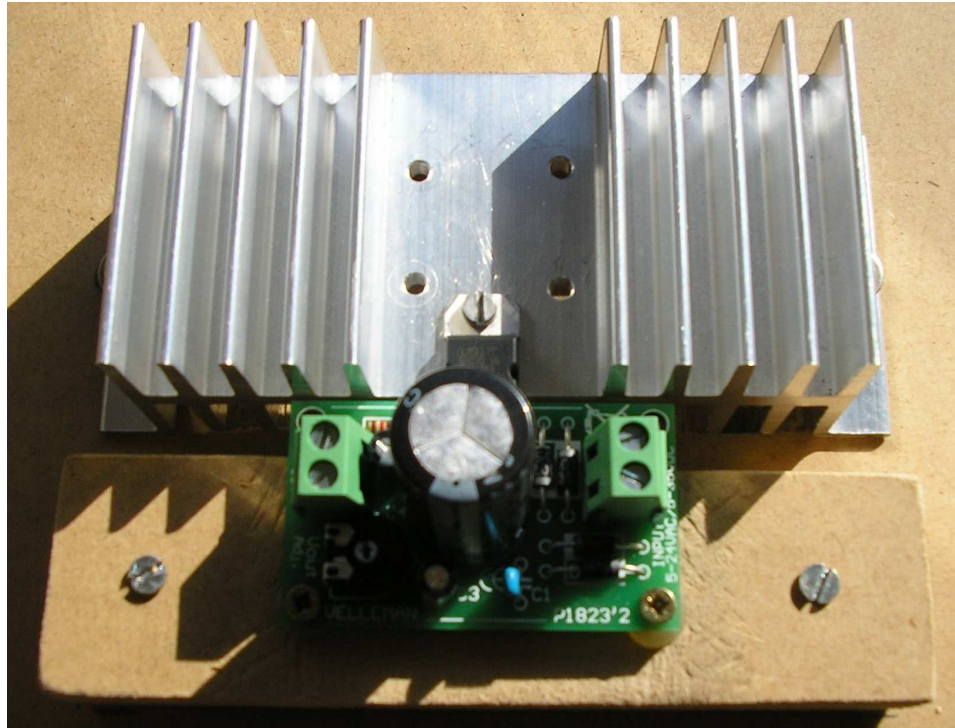
V modelu domu jsou potřeba dvě různá napětí: 12 V a 5 V. 12 V napájení je potřeba pro ventilátor, topení a pro napájení Arduino kitu. 5 V napětí je nutné pro napájení desky se spínací elektronikou, která sama napájí zbytek všech akčních prvků.

#### 7.3.1 Napájení 12V

Celý model je napájen spínaným zdrojem o napětí 12 V. Maximální možný proud zdroje je 2250 mA. Toto napětí je v modelu dovedeno na svorkovnici, odkud je „čerpáno“ do všech prvků, které jsou na 12 V.

#### 7.3.2 Napájení 5V

Pro získání 5 V napájecí větve je v modelu použit stabilizovaný napájecí zdroj F-KVM124. Ten má na vstupu 12 V, které jsou do něj dovedeny z hlavního napájecího zdroje. Na výstupu má napětí 5 V a maximální proud 1000 mA. Integrovaný stabilizátor nemá vlastní chladič, ten bylo potřeba přimontovat dodatečně. Celý tento zdroj i s pasivním chladičem je v modelu umístěn v zadní části podstavce, a to zevnitř na odnímatelném krycím panelu. V panelu je obdélníkový otvor pro přístup vnějšího vzduchu k pasivu.



Obrázek 27: 5 V zdroj stabilizovaného napětí

## 7.4 Elektroinstalační materiál

### 7.4.1 Kabeláž

Pro veškeré elektrické rozvody byly použity vodiče typu CYA 0,35. Jedná se o vodič s měděným lankem o průřezu  $0,35 \text{ mm}^2$  a PVC izolací. Celkem je v modelu domu více než 43 m tohoto drátu.

### 7.4.2 Další použitý materiál

Vodiče vedoucí do svorkovnic a do desky se spínací elektronikou jsou zakončovány dutinkami o délce 8 mm a průřezu  $0,5 \text{ mm}^2$ . Jedná se o dutinky s izolací. Vodiče vedoucí do předního panelu s tlačítky jsou zakončovány konektory typu FASTON.

## 8 SOFTWARE

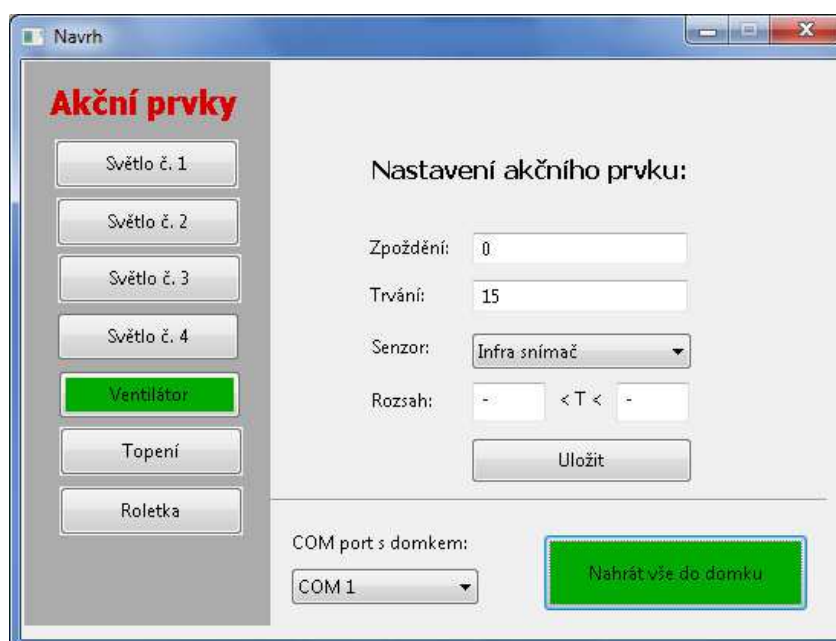
Software pro model domu se skládá ze dvou částí: software do mikropočítače a software do osobního počítače PC, odkud se budou provádět „nastavování“ inteligentního domu. Tyto dva programy spolu komunikují přes USB.

### 8.1 Řídící aplikace na PC

Řídící aplikace v PC je klasická okenní aplikace pro systém Windows. Aplikace byla napsána v jazyce C++. Při tvorbě grafického uživatelského rozhraní byla použita knihovna wxWidgets pro C++. Celá aplikace byla vytvořena ve vývojovém prostředí wx-DevC++, které má již integrován generátor kódu pro rychlou a snadnou tvorbu okenních aplikací.

#### 8.1.1 Vzhled řídicí aplikace

Aplikace je uživatelsky jednoduchá, jak ilustruje obrázek 28.



Obrázek 28: Vzhled řídicí aplikace na PC

V levé části je sloupeček se seznamem všech sedmi akčních prvků. Po kliknutí na kterýkoliv prvek dojde k jeho zelenému zvýraznění, a v hlavní části okna se objeví nastavení vybraného prvku. Zde si může uživatel vybrat z roletkového menu senzor – tedy

snímač kterým bude vybrán akční prvek aktivován. Dále lze přesně definovat zpoždění, s jakým dojde k aktivaci prvku po „spuštění“ daného snímače. V kolonce „Trvání“ volíme, po jak dlouhou dobu má zůstat akční prvek spuštěn. V případě, že jako vstupní snímač zvolíme některý z teploměrů, má zde smysl nastavení rozsahu teploty, při které dojde ke spuštění akčního prvku.

Stejným způsobem si uživatel může nastavit všechny parametry u každého prvku. Kliknutím na tlačítko „Nahrát vše do domku“ dojde k přenesení veškerého nastavení do modelu. Model od této chvíle začne fungovat podle nastavených parametrů, a to bez závislosti na dalším spojení s počítačem.

### 8.1.2 Třída Prvek

V aplikaci je potřeba nějak uchovávat veškeré „naklikané“ nastavení, dokud nedojde k nahrání do modelu. Po kliknutí na „Nahrát vše do domku“ se posílají přes USB do modelu všechny parametry pro všechny prvky. Pro potřeby ukládání dat je v programu třída „Prvek“. Tato třída v sobě zapouzdřuje veškeré nastavení pro jeden akční prvek. Pro každý akční prvek je v programu vytvořena jedna instance této třídy. Při nastavování konkrétního prvku se zadávané parametry ukládají do konkrétní instance tomuto prvku přidělené. Pro ilustraci zde vkládám kód deklarující třídu Prvek:

```
class Prvek
{
    public:
        Prvek()
        {
            Zpozdeni = 0;
            Trvani = -1;
            Senzor = 4;
            RozsahFrom = 20;
            RozsahTo = 25;
        }

        float Zpozdeni, Trvani;
        int Senzor; /* 1: Teplomer 1 2: Teplomer 2,
                    3: Pohybovy snimac, 4: Infra snimac*/
        float RozsahFrom, RozsahTo;
};
```

Jak je vidět, třída je velmi jednoduchá. Obsahuje pět proměnných, které představují všechny parametry, které se dají jednomu prvku nastavit. Konstruktor třídy nedělá nic jiného, než počáteční nastavení těchto proměnných. Parametr „Trvání“ může nabývat hodnoty -1, což představuje nekonečně dlouhé trvání zapnutí akčního prvku.

### 8.1.3 Sériová komunikace

Pro naprogramování sériové komunikace byla použita knihovna Serial. Tato knihovna je napsaná v jazyce C++ a je k dispozici zdarma. Její použití spočívá ve vytvoření instance třídy CSerial. Nad touto instancí pak můžeme volat některé metody užitečné pro sériovou komunikaci.

Metoda **bool Open( int nPort = 2, int nBaud = 9600 )** slouží k otevření COM portu. Parametr nPort určuje číslo portu, který chceme otevřít, nBaud je požadovaná přenosová rychlost.

Metoda **int SendData( const char \*buffer, int size )** slouží k odeslání dat po sériové lince. Parametr buffer je ukazatel na začátek pole prvků typu char, kde se předpokládá výskyt dat pro odeslání. Parametr size je délka řetězce, který se má odeslat. Metoda vrací počet prvků které byly skutečně přeneseny.

V aplikaci dochází k otevření COM portu až po kliknutí na tlačítko „Odeslat vše do domku“. Následně se načtou všechna data reprezentující parametry akčních prvků, převedou se na řetězec znaků typu char, a tento řetězec je vyslán přes otevřený COM port.

## 8.2 Řídící software pro mikropočítač

Software pro mikropočítač je napsán v prostředí Arduino. Jazyk vychází z jazyka C++ a až na pár drobností jsou tyto jazyky totožné. Prostředí Arduino zároveň umožňuje zkompilovaný program odeslat do vývojového kitu Arduino.

### 8.2.1 Třída snímač

V programu je vytvořena třída snímač, která v sobě zapouzdřuje obecnou funkčnost společnou všem čtyřem snímacím prvkům. Každý snímač má v programu svoji instanci této třídy.

```
class Snimac
{
    private:
        int pin;
        boolean type; // TRUE = digital, FALSE = analog

    public:
        Snimac(int p, boolean digital = true)
        {
            type = digital;
            pin = p;
            pinMode(p, INPUT);
        }

        float GetValue();
        boolean GetType(){return type;};
};
```

Jak je vidět v deklaraci třídy „Snimac“, obsahuje dvě privátní proměnné. Proměnná „pin“ je určena pro číslo pinu, na který bude daný snímač připojen. Pravdivostní proměnná „type“ určuje, zda daný snímač poskytuje analogovou nebo digitální informaci. To je potřebné pro metodu **float GetValue()** (zde nerozepsáno její tělo), která má za úkol nezávisle na typu snímače vrátit jeho hodnotu. Tato metoda je napsána tak, že u digitálních snímačů vrátí hodnotu 0 nebo 1, kdežto u analogového snímače vrátí přímo číslo získané z A/D převodníku.

### 8.2.2 Třída Prvek

Podobně jako řídicí okenní aplikace, i program do mikropočítače obsahuje třídu s názvem „Prvek“, obecně reprezentující akční prvek.

```

class Prvek
{
private:
    int pin;
    Snimac* s;
    boolean ON;

public:
    Prvek(int p, Snimac* sn = NULL);

    long zpozdeni, trvani;
    float from, to;
    int eventOn, eventOff;

    Snimac* GetSnimac(){return s;};
    void SetSnimac(Snimac* set_s){s = set_s;};
    boolean GetOn(){return ON;};
    void SetOn(boolean set){ON = set;};

    void KillEvents();
    void Vypnout();
    void Zapnout();
    boolean IsTriggered(); // zda ma dojit k aktivaci prvku
};

```

Třída „Prvek“ je již o něco rozsáhlejší, takže zde popíšeme jen ty nejdůležitější části. Třída obsahuje ukazatel „s“ na objekt třídy „Snimac“. To je důležité, protože tímto ukazatelem se určuje, který snímač je k danému akčnímu prvku přiřazen. Toho využívá například metoda **boolean IsTriggered()**, která kontroluje, zda existuje požadavek na spuštění akčního prvku. Přes tento ukazatel se metoda dostane k snímači, z něhož si „vytáhne“ pomocí jeho metod potřebnou hodnotu. Pro názornost zde uvedu tělo této metody:

```

boolean Prvek::IsTriggered() // zda ma dojit k aktivaci prvku
{
    if( ON || s == NULL ) return false;

    float HodnotaSnimace = s->GetValue();
    if( s->GetType() )
    {
        // digitalni
        if( HodnotaSnimace == 0 ) return true; else return false;
    }
    else
    {
        // analogovy
        if( from < HodnotaSnimace && HodnotaSnimace < to ) return true;
        else return false;
    }
}

```

### 8.2.3 Hlavní smyčka programu

Zde je uveden kód hlavní smyčky programu:

```
void loop()
{
  // kontrola prepnutí režimu
  if( digitalRead(pinRezim) == LOW && manual && PrepinaniPovoleno )
    Automat();

  if( digitalRead(pinRezim) == HIGH && (!manual) && PrepinaniPovoleno )
    Manual();

  if(!manual)
  {
    // kontrola ventilatoru:
    if( Ventilator->IsTriggered() )
    {
      //Serial.println("Triggered");
      Ventilator->eventOn = t.after(Ventilator->zpozdeni,VentilatorStart);
      Ventilator->SetOn(true);
    }
    //.
    // Stejne tak kontroly pro ostatni prvky
    //.
  }

  t.update();
}
```

Na začátku smyčky se provede kontrola, zda došlo k požadavku na přepnutí režimu (manuálním přepínačem „pinRezim“). Pokud ano, zavolá se příslušná pomocná funkce Automat() nebo Manual(). Tyto funkce provedou všechny nezbytné kroky, které jsou pro přepnutí režimu nutné. Následně se program rozvětví podle aktuálního režimu, který je signalizován pravdivostní proměnnou „manual“. V případě manuálního režimu se neděje nic. U automatického režimu ( !manual ) se provedou kontroly spuštění jednotlivých prvků. Ve výše uvedeném příkladu kódu je pro jednoduchost uvedena pouze kontrola ventilátoru.

### 8.2.4 Časování

Pro programování časových zpoždění jsou u Arduina k dispozici funkce jako například `delay()`, která program pozastaví na zadaný počet milisekund. Problémem této funkce je, že během tohoto čekání je celý program „zamrzlý“, a během samotného čekání tedy nemůže mikro počítač provádět nic jiného. To je velký problém, protože program v mikro počítači musí být schopen zachytit příchozí sériovou komunikaci. Pokud by komunikace přišla zrovna v okamžiku, kdy program na něco čeká, s největší pravděpodobností by komunikaci nezachytil. Sériová linka mikro počítače sice ukládá příchozí bajty do vyrovnávacího bufferu, jenže ten má jen 64 bajtů a můj komunikační protokol obsahuje kolem dvou set bajtů. Z těchto důvodů nesmí být nikde v programu čekání, které by program byl jen na chvíli „zamrazilo“. [2]

Veškeré čekací a jiné časovací záležitosti byly vyřešeny pomocí knihovnic funkcí knihovny `Timer`. Tato knihovna umožňuje vytvářet časovací události, jejich odpočítávání se děje na pozadí programu. V hlavní smyčce stačí volat nad vytvořenou instancí třídy `Timer` metodu `update()`, jak je vidět na ukázce kódu na předchozí straně.

## ZÁVĚR

Cílem této práce bylo vytvořit funkční model inteligentního domu. K tomu bylo nejprve potřeba seznámit se s problematikou inteligentních budov. Ta byla podrobně nastudována a v teoretické části práce je popsána. Byly uvedeny některé nezákladnější definice pojmů jako jsou inteligentní budova, inteligentní dům.

V rámci praktické části pak byl navržen a vyroben funkční model inteligentního domu.

Pro model domu byly navrženy funkce vhodné pro demonstraci možností ovládání a regulace v moderním domě, jako jsou elektrické topení, elektrická roleta, ventilátor a světla. Všechny tyto akční prvky mohou být na základě naprogramovaných funkcí nebo manuálních tlačítek aktivovány.

Pro ovládání funkcí modelu domu bylo navrženo a realizováno vhodné rozhraní. Funkce lze ovládat buď přímo pomocí manuálních přepínačů umístěných na ovládacím panelu, nebo automaticky na základě programu využívajícího předem zadané parametry. Pro možnost úpravy těchto parametrů uživatelem modelu bylo vytvořeno softwarové rozhraní, které běží na připojeném osobním počítači a s modelem komunikuje přes USB kabel.

Výsledkem práce je reálný dřevěný model domu, ve kterém byly realizovány všechny navrhnuté funkce, včetně manuálního ovládacího panelu a možnosti připojení k PC s řídicí aplikací.

Pro řízení modelu domu byl využit mikropočítač. Pro tento mikropočítač bylo vytvořeno programové vybavení za použití vývojového prostředí Arduino.

V rámci práce také vzniklo programové vybavení pro osobní počítač tak, aby bylo možné jej využít k ovládání modelu domu a k úpravě jeho funkcí. Toto programové vybavení bylo vytvořeno v jazyce C++. Při tvorbě byla využita knihovna wxWidgets.

Model domu je v současné době ve fázi dokončování a testování. Následně bude předán pro účely Experimentária do Otrokovic.

**SEZNAM POUŽITÉ LITERATURY**

- [1] BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0
- [2] CATSOULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3
- [3] MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6
- [4] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1
- [5] VALEŠ, Miroslav. Inteligentní dům. 1. vyd. Brno: ERA, 123 s., il. (část barev.). ISBN 80-736-6062-8
- [6] LTD, DTD, MDF: co znamenají záhadné zkratky na nábytku?. In: Bydlení [online]. 2008 [cit. 2014-05-12]. Dostupné z: [http://bydleni.idnes.cz/ltd-dtd-mdf-co-znamenaji-zahadne-zkratky-na-nabytku-f8i-/stavba.aspx?c=A080522\\_121457\\_dum\\_stavime\\_web](http://bydleni.idnes.cz/ltd-dtd-mdf-co-znamenaji-zahadne-zkratky-na-nabytku-f8i-/stavba.aspx?c=A080522_121457_dum_stavime_web)
- [7] PIR detektor: skvělý sluha, ale zlý pán. In: Hw.cz [online]. 2013 [cit. 2014-05-12]. Dostupné z: <http://www.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>
- [8] LM35 Precision Centigrade Temperature Sensors. Texas, 2013.
- [9] ARDUINO [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www.arduino.cc/>
- [10] Inteligentní budovy [online]. 2010 [cit. 2014-05-14]. Dostupné z: <http://www.inteligentni-budovy.cz/>
- [11] Mistrovství v C++. Brno: Computer Press, 2004. ISBN 80-251-0098-7.
- [12] Wiring [online]. 1999 [cit. 2014-05-14]. Dostupné z: <http://wiring.org.co/>
- [13] Eagle Online [online]. 2014 [cit. 2014-05-14]. Dostupné z: <http://www.eagle.cz/>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

USB	Univerzální sériová sběrnice.
OOP	Objektově orientované programování.
API	Rozhraní pro programování aplikací.
DPS	Deska plošných spojů.
PCB	Deska plošných spojů (anglicky).
AD	Analogově digitální převodník.
FLASH	Elektricky programovatelná paměť s libovolným přístupem.
EEPROM	Elektricky mazatelná paměť jen pro čtení.
SRAM	Statická paměť.
LTD	Laminované dřevotřískové desky.
MDF	Dřevovláknitá deska.
LED	Světlo emitující dioda.
PIR	Pyroelektrický snímač.
FET	Unipolární tranzistor.
PVC	Polyvinylchlorid.

**SEZNAM OBRÁZKŮ**

Obrázek 1: Řadový akční člen KNX/EIB. Řídí osm nezávislých žaluziových pohonů s možností ručního ovládání _____	13
Obrázek 2: Náhled na vývojové prostředí Arduino _____	16
Obrázek 3: Náhled na editor schémat v systému EAGLE _____	18
Obrázek 4: Vývojový kit Arduino Uno _____	19
Obrázek 5: Návrh a rozmístění místností a funkčních prvků _____	22
Obrázek 6: Nárýs, půdorys a bokorys modelu domu _____	25
Obrázek 7: Nárýs, půdorys a bokorys podstavce _____	26
Obrázek 8: Blokové schéma elektroniky v modelu _____	27
Obrázek 9: Světlo v obývacím pokoji, tvořené třemi LED diodami _____	28
Obrázek 10: Ventilátor umístěný v kuchyni _____	28
Obrázek 11: Výkonové rezistory použité jako topné těleso _____	29
Obrázek 12: Motorek s pákovým mechanismem pro vytahování roletky – umístěno v podstavci _____	30
Obrázek 13: Blokové schéma snímače teploty LM35 _____	31
Obrázek 14: Blokové schéma PIR detektoru _____	31
Obrázek 15: Schéma zapojení spínací elektroniky pro spínání světel _____	33
Obrázek 16: Schéma zapojení spínání ventilátoru _____	34
Obrázek 17: Schéma zapojení spínacího obvodu pro relé _____	34
Obrázek 18: Schéma zapojení obvodu pro spínání motorku elektrické roletky _____	36
Obrázek 19: Kombinační obvod podle návrhu _____	38
Obrázek 20: Optimalizovaný kombinační obvod _____	39
Obrázek 21: Schéma zapojení obvodu pro volbu spínacího režimu _____	40
Obrázek 22: Model domu po sestavení a slepení hotových dílů _____	41

---

Obrázek 23: Navržená deska plošného spoje – pohled ze strany součástek_____	42
Obrázek 24: Zobrazení drátových propojek_____	43
Obrázek 25: Předloha (maska) pro výrobu plošného spoje vytištěná na průhledné folii__	44
Obrázek 26: Hotová a osazená deska_____	45
Obrázek 27: 5 V zdroj stabilizovaného napětí_____	47
Obrázek 28: Vzhled řídicí aplikace na PC_____	48

## SEZNAM TABULEK

Tabulka 1: Pravdivostní tabulka kombinačního logického obvodu pro řízení roletky\_\_\_\_37

## SEZNAM PŘÍLOH

- P I Dokumentační CD obsahující elektronickou verzi této diplomové práce, řídicí aplikaci pro PC včetně zdrojových souborů a zdrojový kód programu do mikropočítače.
- P II Elektrické schéma desky se spínací elektronikou.

# PŘÍLOHA P II: ELEKTRICKÉ SCHEMA DESKY SE SPÍNACÍ ELEKTRONIKOU

