



Středoškolská technika 2017

Setkání a prezentace prací středoškolských studentů na ČVUT

Model Inteligentního domu

Honzík Jan

SPŠ A VOŠ PÍSEK

Karla Čapka 402, 397 11 Písek

Anotace:

Práce řeší návrh a konstrukci funkčního modelu inteligentního domu. Model se skládá z arduino UNO, ethernetového štítu, routeru a modelu domu s deskou plošného spoje, která tvoří elektroinstalaci modelu. Arduino lze napájet stejnosměrným napětím 5V nebo 9V. Výkonové prvky tvoří LED diody. Díky inteligentní instalaci zatím lze ovládat osm světel a topení. Ovládacími prvky jsou tlačítka a zároveň webová aplikace, která je spuštěná na serveru. Server je naprogramován přes arduino, které je pomocí ethernetového štítu připojeno k síti.

Annotation:

Graduation work deals with the design and construction of the functional model of the intelligent house. This model consists of Arduino Uno, Ethernet shield, router and a model house with a printed circuit board, which forms an electro installation of the model. Arduino can be powered by a DC voltage of 5V or 9V. Circuit elements are LEDs. Thanks to the installation of smart home eight lights and heating can be controlled. The controls are buttons and a Web application that is running on the server. This server is performed by Arduino, which is connected via an Ethernet network shield.

Klíčová slova:

Arduino UNO, Ethernet štít, LED dioda, rezistor, inteligentní dům, webová aplikace, elektroinstalace.

Obsah

Anotace	4
Úvod	5
1 Teoretická část.....	5
1.1 Základní vlastnosti Arduino UNO	5
1.2 Programovací jazyk.....	6
1.3 Arduino IDE.....	6
1.4 LED Diody.....	7
1.5 Čidlo DHT 11	7
1.6 Router RDAA 81.....	7
1.7 Posuvný registr 74HC595	8
2 Praktická část.....	9
2.1 Současné možnosti inteligentních instalací.....	9
2.1.1 Funkce inteligentních domů	9
2.1.2 Nabídka na trhu.....	10
2.1.3 Novinky v chytrých domácnostech.....	11
2.1.4 Cena instalace.....	11
2.2 Realizace modelu domu	12
2.2.1 Vytváření návrhu plošného spoje v programu Eagle.....	13
2.2.2 Tvorba plošného spoje	13
2.2.3 Zhotovení 3D modelu	14
2.2.4 Rozmístění jednotlivých prvků	15
2.3 Návrh systému.....	15
2.3.1 Připojení systému k Arduino UNO.....	16
2.4 Webové rozhraní	16
2.4.1 Ethernet štít a server	16
2.4.2 Připojení do domácí sítě.....	17
2.4.3 Připojení k internetové síti	17
2.4.4 Ovládání z webového rozhraní.....	17
2.5 Ovládání komponentů	18
2.6 Napojení na výkonové prvky a spínací součástky.....	20
2.6.1 Tlačítková klávesnice	20
2.6.2 LED diody připojené na posuvný registr.....	21
2.7 Testování systému.....	21
2.8 Možné vylepšení modelu	22
Závěr	23

Zdroje	25
Přílohy:.....	26

Úvod

V této práci řeším návrh a realizaci modelu inteligentního domu. K realizaci jsem si vybral mikropočítač, který je totožný s Arduino UNO a funguje na procesoru ATmega328. Jako výkonové prvky jsou použity LED diody, které představují rozsvícení žárovky, a nebo spuštěné topení. Model domu tvoří desky plošného spoje, které lze libovolně přidat, nebo odebrat. Mým úkolem je vytvořit program, který bude schopen přijímat pokyny z hardwaru a zároveň z webové aplikace inteligentního domu pro vzdálené ovládání.

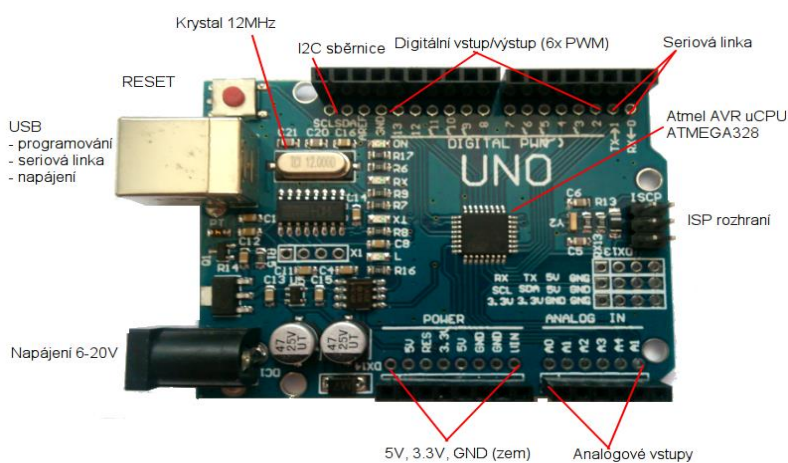
Pro svou dlouhodobou maturitu jsem si vybral toto téma, protože mě fascinuje inteligentní instalace. A jednou bych chtěl použít získané dovednosti ve své domácnosti. Proto jsem zvolil ovládání přes zmíněný mikropočítač, protože pořizovací cena je přibližně 150Kč a také proto, protože mikropočítače na bázi arduina mě baví programovat.

1 Teoretická část

1.1 Základní vlastnosti Arduino UNO

Arduino je otevřená platforma s grafickým vývojovým prostředím, které vychází z prostředí Wiring a Processing. Může být použito k vytváření samostatných interaktivních zapojení nebo může být připojeno k software na počítači (např. Adobe Flash, Processing, Max/MSP, Pure Data...). Řídicí program je vyvíjen zvlášť (na stolním počítači) a do Arduina je posléze nahrán a spuštěn. Uvnitř Arduina je pak spuštěn jen tento program, který typicky obsahuje smyčku, která se neustále opakuje (Arduino neustále zjišťuje stav svého okolí a na změny reaguje). Díky tomu má nízkou spotřebu (je možné napájení malou 9V baterií) a hodí se například pro řízení robotů, dronů a podobně. Desky arduino obsahují 8bitové mikrokontrolery z rodiny AVR od firmy Atmel (Atmega328) a množství dalších podpůrných obvodů.

Každá deska má většinu I/O pinů přístupných přes standardizované konektory, do kterých se jednoduše připojují další obvody, kterým se ve světě Arduina říká Shildy(štíty). Na deskách je několik diod, resetovací tlačítko, konektory pro ICSP programování, napájecí konektor, oscilátor a obvod



Obrázek 1: Arduino UNO - popis

zprostředkovávající komunikaci po USB. Arduino Uno, poskytuje celkem 14 I/O digitálních pinů a 6 pinů analogových. Šest digitálních pinů je také možné použít na softwarově řízený PWM výstup. Hlavní mikrokontroler, který je uživatelsky programovatelný, již má bootloader (kód, který se po spuštění postará o základní nastavení mikrokontroleru, jako jsou interní časovače, nastavení rozhraní USART a další) a nastavené potřebné fuses bajty (těmi se nízkourovňově nastavují některé vlastnosti čipu). Díky tomu se uživatel nemusí starat o detaily a své programy píše v jazyce podobném C/C++. Ačkoliv je Arduino připojeno k počítači pomocí rozhraní USB, je softwarově simulována sériová komunikace přes linku RS-232. Parametry: mikroprocesor Atmel ATmega328P, takt CPU 16Mhz, flash paměť 32kB, SRAM paměť 2kB, EEPROM paměť 1Kb.

1.2 Programovací jazyk

Jak jsem již uvedl, tak arduino se programuje pomocí jazyka, který je podobný C/C++ a to je jazyk wiring. Wiring vlastně není úplně programovací jazyk, ale spíše knihovna jazyka C/C++. Program by měl mít strukturu: vložení knihoven (zápis: „#include <Ethernet.h>“ nám vloží knihovnu Ethernet), definice proměnných (např. zápis „int tlacitko = A0“ proměnná tlacitko může nabývat celočíselné hodnoty, která je na analogovém vstupu A0), nastavení vstupů/výstupů a komunikace (např. zápis „pinMode(tlacitko, INPUT_PULLUP)“ tento zápis řekne arduinu, že tlacitko(analogový pin A0) je vstup přes PULLUP rezistor), hlavní program („loop(){ ...}“ obsahuje vlastní (hlavní) program, který se cyklí.) a pomocné funkce, podprogramy.

1.3 Arduino IDE

Arduino má vlastní programovací prostředí, které se jmenuje Arduino IDE. Tento program pracuje v operačních systémech Windows, Linux a iOS. Software obsahuje mnoho prvků. V horní liště programu nalezneme rolovací menu File, kde můžeme vytvořit nový projekt, otevřít starší projekt, otevřít nedávno zavřený projekt, dále se zde nachází pole sketchbook, která nám zobrazí všechny uložené projekty, dalším mnou uvítaným polem je examples, kde se nachází již napsané jednoduché programy a máme zde možnosti uložit/uložit jako. V menu Edit nalezneme mnoho pomocníků při úpravě

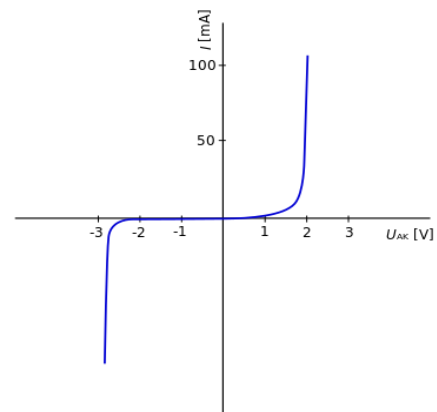


Obrázek 2: Arduino IDE

a opravách programu, pod menu Sketch můžeme překládat, odesílat program do arduina, přidávat a spravovat knihovny. Menu Tools nám umožňuje otevřít serial monitor (vypisuje nám hodnoty z arduina odeslané přes seriálovou linku a to zápisem `Serial.print(ln)` jako další zobrazovací jednotka je seriál plotter, ten nám ukazuje hodnotu napětí na vstupu, který čteme v aktuálním čase. Dále zde můžeme nastavit, jakou desku programujeme a port, přes který komunikujeme s deskou. Pod menu Help se nám zobrazí nápovědy na ovládání programu. Pod touto lištou máme vedle sebe 6 ikon: zleva přeložit program, přeložit a nahrát program do arduina, vytvořit nový projekt, otevřít uložený projekt, uložit projekt a poslední seriál monitor. Následuje textové pole, kde píšeme vlastní program, viz Programování arduina. V černém poli se nám zobrazují hlášení (kde je chyba, apod.).

1.4 LED Diody

LED neboli světlo imitující dioda (Light Emitting Diode) je polovodičová elektronická součástka, která se chová obdobně jako obyčejná dioda (viz Obrázek 3), ale emituje elektromagnetické záření (infrachervené, ultrafialové a viditelné světlo). Vlnění vzniká na přechodu P-N, kterým prochází elektrický proud v propustném směru. Vlnová délka elektromagnetického záření závisí na chemickém složení použitého polovodiče. Jas lze regulovat velikostí proudu. Další možností regulace je připojení na zdroj pulzního proudu s určitou frekvencí (pro lidské oko by měla stačit frekvence 30Hz). Tento jev vzniká postupným rozsvěcováním a zhasínáním LED a velikost jasu nám přijde rozdílná podle doby, jak dlouho prochází LED diodou proud a jak dlouho neprochází.



Obrázek 3: Volt - Ampérová charakteristika

1.5 Čidlo DHT 11

Teploměr a zároveň vlhkoměr, který měří teplotu v rozsahu 0°C až 50°C s přesností plus, minus 1°C a vlhkost 20% až 90% s přesností plus, minus 4%. Je to velmi jednoduché čidlo právě pro mikroprocesory jako je právě arduino. Na modulu jsou tři vývody (možno zakoupit i se čtyřmi piny, ale jeden se nezapojuje, takže je to zbytečné). Jeden pin je pro napájení stejnosměrného napětí +5V, další pin je datový a poslední pin se připojí na zem.



Obrázek 4: Čidlo DHT 11

1.6 Router RDAA 81

Pro možnost ovládání z mobilního telefonu, tabletu, a podobně je třeba připojit arduino do sítě. Nalezneme zde tři zdířky pro konektory RJ 45, zdířku pro připojení napájení (stejnoseměrné napětí 5 V, 2 A), resetovací tlačítko a signalizační diody (PWR - zařízení je spuštěné, WLAN – šíří signál wi-fi, 1, 2, 3 – aktivní připojení konektoru RJ 45). Router vysílá dle standard 802-11 a/g tj. ve frekvenčním

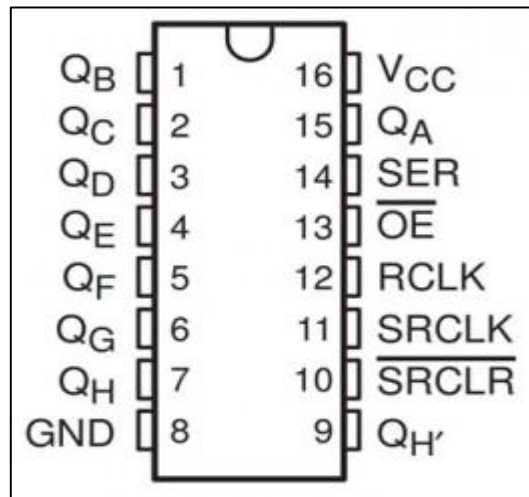


Obrázek 5: Router RDA 81

pásmu 2,4 GHz s maximální rychlostí 54 Mbit/s.

1.7 Posuvný registr 74HC595

Je složen z klopných obvodů, které se starají o posun hodnoty na výstupu o jeden bit. Toto posouvání probíhá díky tzv. hodinovému signálu s určitou frekvencí. Posuvný registr se ovládá pomocí tří vstupů SER pin 14 (zde posíláme data), SRCLK pin 11 (při log „1“ se posunou data), SCLK pin 12 (při log „1“ se data zobrazí na výstupu. Dále je potřeba registr napájet 5 V a to na pinech 16 a 10 a na piny 8 a 13 připojit zem (GND). Pin 9 se nezapojuje. Piny 1-7 a 15 jsou výstupní.



Obrázek 6: Posuvný registr 74HC595

2 Praktická část

2.1 Současné možnosti inteligentních instalací



Obrázek 7: Inteligentní dům - koncept

2.1.1 Funkce inteligentních domů

Cílem inteligentních domů je úplná automatizace a jednoduchost ovládání odkudkoliv. Současné inteligentní instalace se rozvíjejí velmi rychle a možnosti stále přibývají. Mezi hlavní funkce inteligentní instalace patří nastavení světelných scén (jas žárovek, barva RGB pásku). Další funkcí je regulace teploty v každé místnosti dle potřeby. Jedna z hlavních funkcí je šetření energie a to díky automatickým a hlavně efektivním použitím žaluzií, které nejen, že zabraňují vstupu světla, ale i tepla (např. v létě). Zároveň velkou úsporou je větrání, které probíhá přes výměníky, kde se vzduch, který jde z venku, přizpůsobí teplotě vnitřního vzduchu, tím pádem nemáme takové ztráty tepla jako při klasickém větrání. Další výhodou jsou automatické činnosti např. při zpuštění TV se nám ztlumí jas světel, nebo když jdeme spát, tak se zvonek přepne ze zvukového na světelný signál a jakmile nejsme doma, tak se přesměruje na mobilní aplikaci, kde se můžeme s návštěvou domluvit. V rámci inteligentní instalace můžeme ovládat veškeré světla, el. zásuvky, spotřebiče, hlídat spotřebu energií, regulovat teplotu, přehrávat jakoukoliv hudbu, odemkat dveře, otevírat vstupní bránu a garáž, ovládat zavlažování, vytvořit zabezpečení domu, vytvořit různé automatické aplikace např. každý den v 7:00 vám kávovar uvaří kávu a mnoho dalších aplikací. Podle mě skoro vše, na co si vzpomeneme. Vývojáři chytrých domácností mysleli i na seniory a výsledkem je např. koberec, který identifikuje stálý tlak a odešle SOS zprávu. Další inteligentní součástí je zubní kartáček, který během čištění zubů měří tep, tlak a teplotu.

2.1.2 Nabídka na trhu

Loxone tento typ inteligentní instalace nám nabízí velkou škálu inteligence. Inteligenci zde tvoří mini-server, kterým je možné ovládat domácnost přes elektroinstalaci, nebo přes wi-fi (není potřeba provádět novou instalaci, ale nevýhodou je napájení spotřebičů a spínacích prvků). Loxone nám umožňuje ovládat teplotu v každé místnosti, vytvořit zabezpečovací systém, šetřit energie, ovládat ozvučení místností, osvětlení, zavírat okna, ovládat žaluzie, zamykat dveře a mnoho dalších možností. Vše lze ovládat přes mobilní telefon, tablet, počítač, nebo přes chytrá tlačítka, kde



Obrázek 8: Chytré tlačítko

rozhod
uje
na vyko
nání
akce
kam
máčkne
te a
kolikrát
.

Další možnost je inteligentní instalace od firmy **INELS**. Zde je inteligence obdobná jako u Loxone, ale pro mě nejvíce zajímavým je možnost koupit si pouze část inteligentní instalace např. Topení na dálku, Dům pod palcem, Hra světla a podobně. Ovšem cena těchto sad je např. u topení na dálku přibližně 10 000 Kč.

2.1.3 Novinky v chytrých domácnostech

V červenci tohoto roku se zahájí prodej chytrého ovladače Sevenhugs, který dokáže komunikovat s více než 25 000 zařízeními. Ovladač funguje tak, že po místnosti rozmístíme senzory, které nám určují, kde co máme. Díky tomu ovladač po zamíření např. na okno nám zobrazí venkovní teplotu a předpověď, nebo po zamíření na světlo nám umožní ovládat jas světla, nebo ho jen zapnout a vypnout. K tomuto ovladači potřebujeme chytré vybavení domu, které můžeme ovládat jen z domova, takže podle mě by bylo dobré funkci tohoto ovladače propojit s již zavedenou inteligentní instalací, kterou lze ovládat přes aplikace v mobilním telefonu, tabletu, PC, notebooku a nebo TV. V české republice zatím není inteligentní instalace moc rozšířená, protože ceny těchto instalací jsou ve stovkách tisíc a lidé, které nemají v elektronice a luxusu zájem, tak moc ani netuší, že již něco takového existuje. V mém modelu jsem zatím vytvořil ovládání osmi světel, topení a snímání teploty a vlhkosti v jedné místnosti.

Firma Samsung uvedla minulý rok chytrou ledničku, která má na dveřích obrovský displej podporující systém Family Hub, který slouží k tomu, aby zobrazil obsah ledničky. Tento obsah lze vidět pomocí nainstalovaných kamer, které jsou rozmístěny a jejich obrazy tak sloučeny, že získaný obraz vypadá jako prosklený. Ovšem pokud se lednička připojí k chytré domácnosti, tak lze vidět obsah kdekoliv např. v obchodě při nákupu. Tento rok chce Samsung uvést na trh ledničku, kterou je možné ovládat přes USB dongle.

Firma LG uvedla minulý rok na trh vysavač, který sám luxuje, nabíjí se a podobně. Jenomže tento vysavač vylepšili systémem, který když zaznamená pohyb, tak odešle pět fotografií přes server chytré domácnosti.

Společnost Acer nabízí svým zákazníkům IP kamery pro domácí mazlíčky. Tyto kamery umožňují např. dát psovi pamlsek, nebo laserem „prohnat“ kočku po pokoji. Tato IP kamera je vhodná např. pro lidi, kteří nebývají často doma.

1.1.4 Cena instalace

Cena je hlavně závislá na naší rozmanitosti. Navýšení oproti „hloupé“ instalaci je přibližně o 20% dražší. Toto navýšení při stavbě nového domu je téměř zanedbatelné. Ale jak jsem již uvedl, záleží na požadavcích a na zvoleném inteligentním rozhraní. Při vhodném zvolení se nám tyto náklady vrátí za ušetřené energie.

<i>IQ instalace s Arduinem</i>	<i>Cena</i>	<i>IQ osvětlení na trhu</i>	<i>Cena</i>
Arduino UNO klon	150 Kč	iNELS (chytré žárovky 3 ks)	4 943 Kč

Tlačítka 18 ks	90 Kč	Každá další žárovka	1157 Kč (*5 ks)
8 spínacích relé	150 Kč	Topení na dálku iNELS	9269 Kč
Ethernet štít klon	150 Kč		
Posuvný registr 74HC595	10 Kč		
Rezistory 20 ks	30 Kč		
Router	500 Kč		
Celkem	1080 Kč	Celkem	19 997 Kč

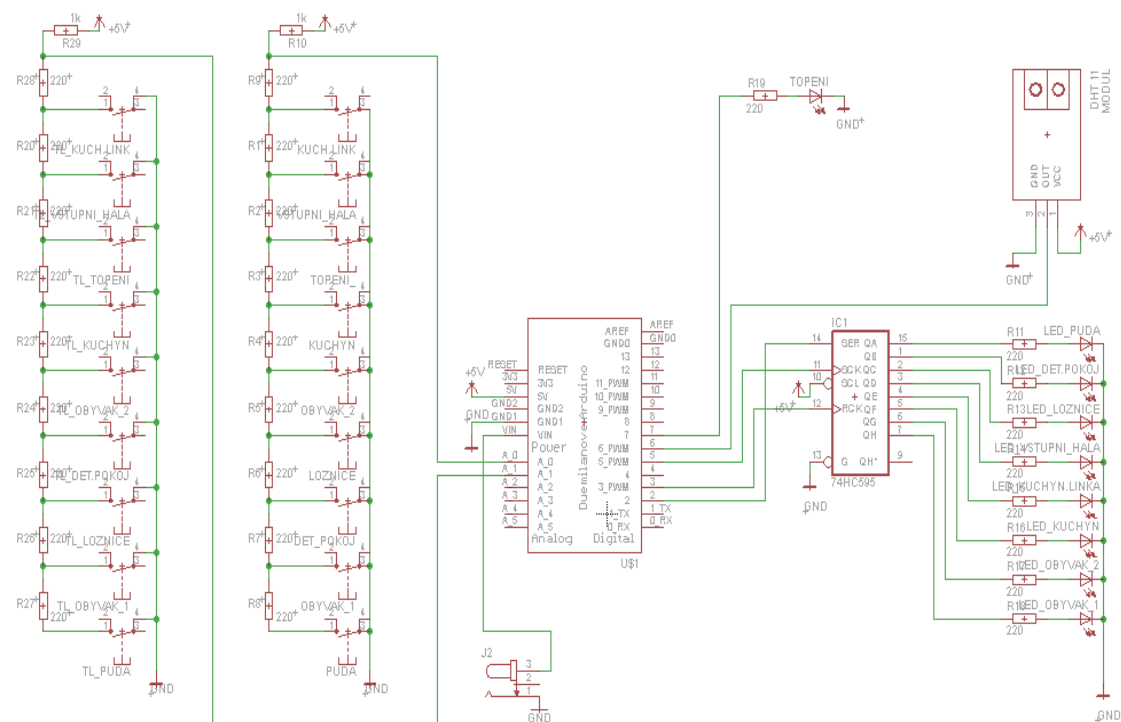
V tabulce jsem započítal náklady na inteligentní instalaci, kterou jsem zatím vytvořil na arduinu. Uvedená cena celkem je cena za rozšíření obyčejné instalace o inteligenci.

Výhody IQ domu s Arduinem: Cena, jednoduchost, rozšiřitelný systém.

Výhody IQ s iNELS: lze upravovat žárovkám jas a barvu, pouze se nainstaluje hardware.

2.2 Realizace modelu domu

Model domu jsem chtěl původně vytvořit pouze ve formátu 2D. Po konzultaci s panem Mgr. M. Janouškem jsem se ale rozhodl, že model vytvořím ve formátu 3D. Model domu je realizován



Obrázek 9: Schéma elektroinstalace modelu

podle schématu pod tímto textem.

2.2.1 Vytváření návrhu plošného spoje v programu Eagle

V tomto programu jsem vytvořil elektrické schéma a návrh desky plošného spoje (viz příloha). Schéma se vytvářel tak, že jsem vložil požadované součástky. U součástek jsem musel dbát na rozměr pouzdra, aby nedocházelo k problémům při osazování. Pro můj model jsem musel vložit do programu knihovny, které obsahují arduino a jeho komponenty. Po vložení součástek jsem navolil jejich hodnotu a název. Následně jsem součástky propojil vodičem. Případné uzly (spojení tří a více vodičů) jsem vytvořil ikonou „Junction“. Názvy a hodnoty součástek by měli být čitelné z jednoho pohledu. Toho jsem docílil ikonou „Smash“ a následným otočením nápisu (ikona „Move“). Po vytvoření schématu jsem mohl začít tvorbu návrhu plošného spoje. (Zde jsem pracoval s reálným vzhledem součástek.) Tyto součásti jsem poskládal tak, abych je mohl co nejjednodušeji spojit a nemusel vytvářet vodivé mosty. Vodivé cesty jsem vytvořil pomocí ikony „Route“. Pro zvětšení díry pro vývody součástek jsem používal ikonu „Via“. Po dosažení optimálního návrhu desky plošného spoje jsem tento návrh vyexportoval ve formě obrázku, který se vytiskl.



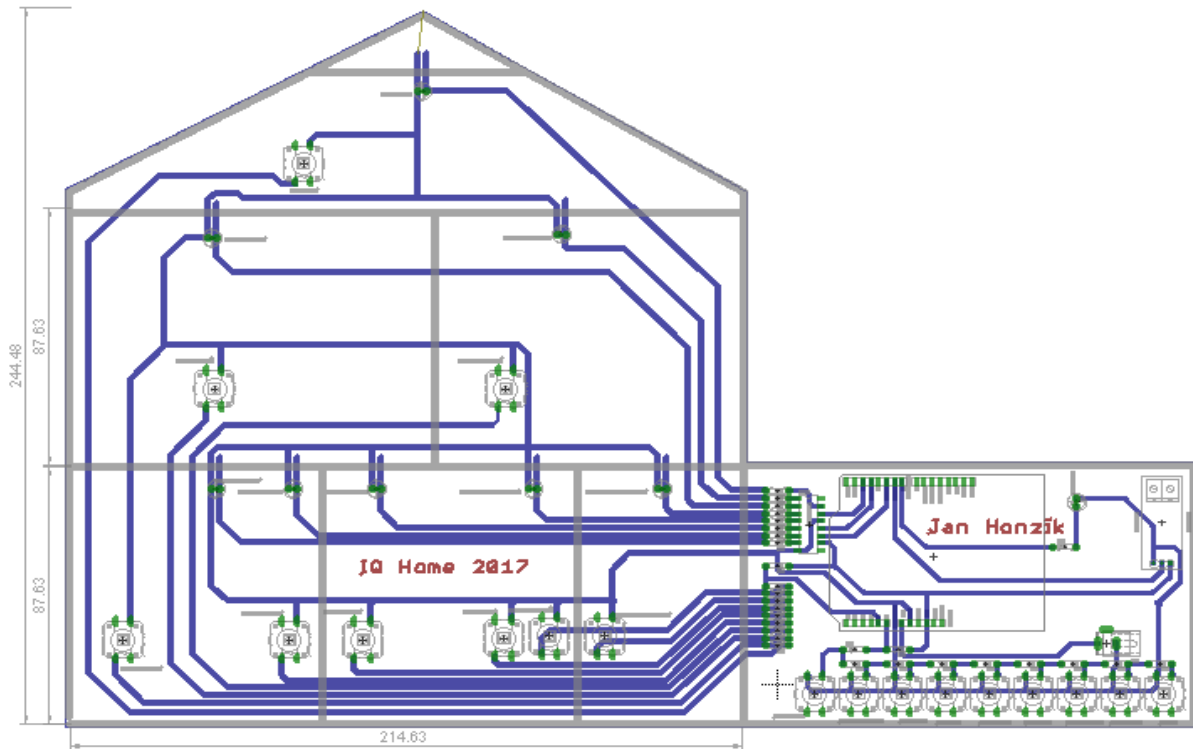
Obrázek 10:
Ovládací prvky
softwaru Eagle

2.2.2 Tvorba plošného spoje

Plošný spoj se používá pro elektrické zapojení elektrických součástek. Využil jsem zde vodivých cest, které se leptají do měděné fólie, která je nanesena na izolační laminátové desce. Tyto vodivé cesty se získají vytvořením schématu v programu EAGLE. V schématu bylo nutné dodržovat velikost pouzdra součástky, aby nedošlo k problémům při osazování. Po vytvoření schématu jsem poskládal součástky tak, aby se nějak nenarušovali a nebylo potřeba použít vnější propojení drátkem (červená vodivá cesta). Po umístění součástek jsem vytvořil vodivé cesty (modrou čarou). V programu je možné si zvolit šířku vodivé cesty od 0,01 palce (0,025 cm) do 0,254 palců (0,645 cm). Další vlastnost programu je zvětšení děr pro vývody součástek. Zvětšením těchto děr jsem zmenšil šanci při vrtání na porušení vodivé cesty a zároveň si zjednodušil práci při pájení.

Jakmile jsem dosáhnul optimálního zapojení součástek, tak jsem návrh vytisknul v poměru 1:1. Vytisknutý návrh jsem nechal vysvítit na desku, která musí být dokonale čistá a odmaštěná tzn., že jsem musel desku dokonale vyčistit pod tekoucí vodou za pomoci jemného smirkového papíru. Po dosažení dokonalé čistoty (vrstva byla jednolitá a byl v ní vidět odraz) jsem desku omyl ještě mýdlem, abych odstranil mastnoty. Před osvětlením je ještě třeba na desku nanést světlo-citlivou vrstvu, která je odolná leptání tzn., osvětlená část se nevyleptá. Po nanesení této vrstvy a jejím roztečením na

desce se deska vloží do pece, kde se nanosená světlo-citlivá vrstva zapeče. Vysvícení probíhá pomocí zářivky v temné komoře, kde může být maximálně světlo z obyčejné žárovky. Po vysvícení se tato deska vyleptá v kyselině např. chlorid železnatý. Po vyleptání je vhodné desku zkontrolovat a



Obrázek 11: Návrh plošného spoje

případně nevyleptané části odstranit. Po kontrole desky se mohou vyvrtat díry pro vývody součástek. Následně osadit a připájet součástky. Po osazení a připájení součástek je třeba vyzkoušet, zda není někde nevodivý tzv. studený spoj. Tato kontrola se dělá nejlépe multi-metrem, nebo ohm metrem.

2.2.3 Zhotovení 3D modelu

Po vyleptání desek plošného spoje a jejich vyzkoušení, jsem z nich vytvořil 3D model. 3D efektu jsem dosáhnul poskládáním a spojením desek a navázáním vodivých cest. Téměř celá vnitřní část modelu je připojena na zemní pin arduina (GND). Díky této vlastnosti jsem využil pro spojení cín, kterým jsem zemnění propojil a zároveň desky připevnil k sobě. Pájením jsem propojil i vodivé cesty, které vodí elektrické napětí a proud do výkonových součástek.

2.2.4 Rozmístění jednotlivých prvků



Obrázek 12: Model IQ domu + popis

2.3 Návrh systému

Systém jsem musel navrhnout tak, aby byl modulární a rozšiřitelný. Model domu lze rozšířit pomocí vyvrtání děr do desky pro vodiče a požadované zařízení. Následně lze zařízení připojit. Model domu lze rozšířit i o další desky plošného spoje a následným propojením s hlavním plošným spojem, který simuluje elektroinstalaci. Systém je možné rozšířit o ovládání žaluzií, zamykání a odemykání dveří, otevírání a zavírání garáže, jednoduchý zabezpečovací systém a mnoho dalších funkcí.

2.3.1 Připojení systému k Arduino UNO

Inteligence této domácnosti je provedena ovládáním osmi světel (ovládaná pomocí posuvného registru **74HC595**, který se řídí **digitálními piny 2, 3 a 5**) a topení (**digitální pin 7**), které znázorňují diody. Ovládat tyto prvky bude možné přes tlačítka v dané místnosti (**analogový vstup A1**), nebo z řídicí místnosti (**analogový vstup A0**) a také pomocí webové aplikace (**IP: 192.168.1.97**). Další prvek bude snímač teploty a vlhkosti a to čidlo **DHT11 (digitální pin 6)**. Teplotu a vlhkost zjistíme právě na webové aplikaci, nebo po připojení počítače k arduinu a spuštěním sériového monitoru v aplikaci Arduino IDE. Na arduinu běží server, díky kterému můžeme ovládat prvky z webového prohlížeče. Arduino je připojeno k síti pomocí Ethernet štítu (**digitální piny 10 – 13**).

2.4 Webové rozhraní

Webové rozhraní je v mém systému nutné k ovládání domu přes síť a to buď domácí, nebo internetovou. Arduino UNO tuto možnost poskytuje, ale je nutné připojit Ethernet štít, vypsát zdrojový kód stránky do arduina a arduino připojit do sítě např. do routeru.



2.4.1 Ethernet štít a server

Ethernet štít je takový štít (nástavba) arduina, díky kterému je možné komunikovat přes ethernetový kabel s okolním světem. K internetu, nebo k domácí síti lze díky tomuto štítu připojit arduino přes konektor RJ45. Tento štít pracuje s Ethernet chipem Wiznet W5100. Ovládání probíhá přes rozhraní SPI. Nalezneme zde slot pro SD paměťovou kartu, resetovací tlačítko a několik diod, pro kontrolu funkčnosti např. RX a TX (odesílání a přijímání dat). Přes ethernetový štít můžeme vytvořit z arduina klienta, nebo server.

Server jsem naprogramoval tak, že přijme dotaz a odešle odpověď. Tato komunikace probíhá odesíláním a přijímáním textu přes HTTP protokol, který pak arduino zpracuje a vyhodnotí. V arduinu, jestliže je serverem, tak je vytvořená i webová aplikace, která umožňuje odesílat příkazy, zobrazovat a číst data z arduina. Webová aplikace v arduinu je jednoduše napsaná, protože zabírá velké

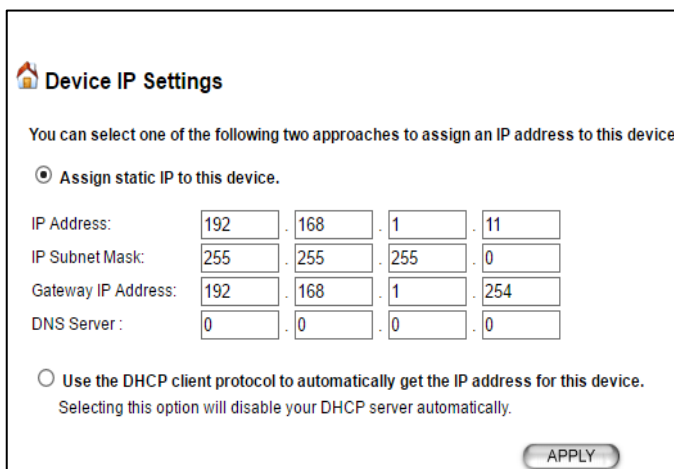
```
// webova komunikace
boolean zacatekCteni = false;
//zač
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 97);
EthernetServer server = EthernetServer(80);
```

Obrázek 14: Zadáání IP a MAC adresy

množství dynamické paměti. Je možné napsat webovou aplikaci na paměťovou kartu, kde by mohla být velmi rozsáhlá, ale statická. Pro naprogramování serveru je potřeba přidat knihovny (SPI.h a Ethernet.h). Další podmínka spočívá v nastavení MAC adresy, která je na originálním štítu uvedena na spodní straně a IP, kterou v domácí síti nastavíme libovolně, nebo při připojení k internetu poskytovatelem tzv. statickou IP (pokud nám poskytovatel není schopný přiřadit statickou IP, existují organizace, které tento problém řeší přidělením vlastního hostitelského jména. Data se odesílají pomocí příkazu client.print/println (pomocí tohoto příkazu odesílám zdrojový kód stránky) a čtení dat se provádí příkazem client.read (tímto příkazem získává arduino příkazy z webové aplikace).

2.4.2 Připojení do domácí sítě

Ethernetovým kabelem jsem musel nastavit vlastnosti routeru. Nastavení se provádí v internetovém prohlížeči po zadání adresy 192.168.1.1 (prvotní adresa routeru, při první instalaci, nebo po restartování). Tuto adresu lze změnit na dynamickou (mění se), nebo statickou (nastavíme vlastní



Device IP Settings

You can select one of the following two approaches to assign an IP address to this device.

Assign static IP to this device.

IP Address: 192 . 168 . 1 . 11

IP Subnet Mask: 255 . 255 . 255 . 0

Gateway IP Address: 192 . 168 . 1 . 254

DNS Server : 0 . 0 . 0 . 0

Use the DHCP client protocol to automatically get the IP address for this device.
Selecting this option will disable your DHCP server automatically.

APPLY

adresu routeru). Pro připojení arduina je nutné nastavit statickou adresu. IP adresa

Obrázek 15: Nastavení routeru

(v protokolu IP verze 4) obsahuje 4 čísla, každé číslo musí nabývat hodnot od 0 do 254. Pro funkční komunikace je potřeba, aby se poslední číslo IP routeru a Ethernet štítu neshodovalo, první tři musí být totožné, tzn. router 192.168.1.11 a Ethernet štít 192.168.1.97. Dále je třeba nastavit název routeru a zabezpečení.

2.4.3 Připojení k internetové síti

Pro ovládání z internetové sítě je potřebné zařídit si u poskytovatele statickou IP, abychom nemuseli při každém odpojení od sítě zjišťovat IP a přeprogramovat arduino. V případě, že poskytovatel by nebyl schopný nám zřídit statickou IP, tak existují servery, které poskytnou vlastní statické hostitelské jméno, a jejich software hlídá změnu naší IP a automaticky nás na ni přesměruje. Pro tuto možnost je potřeba, aby router podporoval DDNS služby. Toto připojení jsem nezkoušel, protože můj router nepodporuje DDNS služby.

2.4.4 Ovládání z webového rozhraní

Ovládání z webové aplikace jsem řešil tak, že po stisknutí na zapnutí nebo vypnutí některého ze světel, nebo topení nás webová aplikace přeměruje na adresu arduino a připiše k ní dvoučíselnou hodnotu (první číslo tj. 0 – 9 nám udává, jaký prvek ovládáme a druhé číslo tj. 1 nebo 0, které určuje, jestli je zařízení zapnuto/vypnuto). Server, který pracuje na arduinou každý cyklus tuto hodnotu čte a dále ji převede z formátu „char“ na „int“. S hodnotou, kterou arduino získalo, provede zkoušku a to který prvek ovládáme a jakou akci chceme vykonat. Po získání této informace přepiše hodnotu v tabulce a ta se objeví na výstupu. V aplikaci vidím pouze zařízení a možnost vypnout/zapnout. Dále zde zjistím teplotu a vlhkost vzduchu a to čidlem DHT 11.

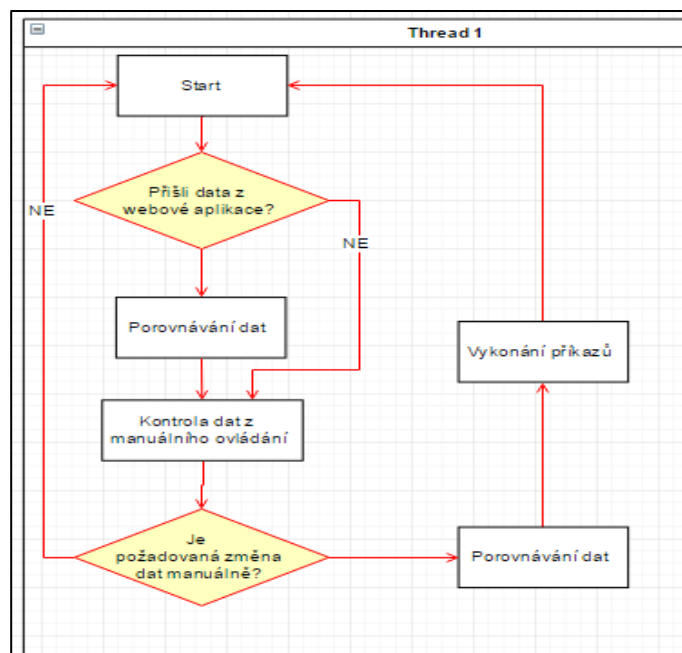
[Zjistit stav domu](#)
 Puda [VYPNI](#)
 Detsky pokoj [VYPNI](#)
 Loznice [VYPNI](#)
 Vstupni hala [ZAPNI](#)
 Kuchynska linka [ZAPNI](#)
 Kuchyn [VYPNI](#)
 Obyvak 1 [ZAPNI](#)
 Obyvakk 2 [VYPNI](#)

[Vypnout vsechny svetla](#)
 Teplota je: 22 *C
[VYPNI Topeni](#)
 Vlhkost je: 78 %

Obrázek 16: Vzhled webové aplikace

2.5 Ovládání komponentů

Arduino se programuje v jazyce C/C++. Základní znalosti jsem získal při výuce ICT ve třetím ročníku. Další znalosti jsem získal samostudiem z webů zaměřujících se na arduino a jeho programování. Znalosti jsem postupně navyšoval a tím pádem jsem i rozšiřoval program pro inteligentní domácnost. Program jsem postupně zkoušel na kontaktním nepájivém poli. Po několika verzích programu jsem dospěl k optimálnímu řešení programu, viz příloha. Nejtěžší bylo zprostředkovat



Obrázek 17: Vývojový diagram programu

komunikaci mezi arduinem a Ethernet štítem, protože se nejedná o originální desky Arduino, ale o čínské klony, které jsou cenově dostupnější. Další významnou část programu tvoří webová aplikace, která běží na arduino serveru. Tato aplikace nám umožňuje ovládat výstupy arduino a číst hodnoty na vstupech. V programu jsou použity knihovny: Ethernet a SPI, díky kterým je snadnější naprogramovat arduino server a komunikaci přes domácí/internetovou síť. Pro chod serveru bylo nutné nastavit MAC adresu a IP adresu serveru (Ethernet štítu s arduinem). Dále specifikovat serverový port a spustit komunikaci mezi serverem a lokální sítí. Poté jsem již mohl odeslat kód pro html webovou aplikaci a zabývat se čtením dat. Data, které jsem četl, byli typu char, pro snadnější

práci jsem tyto data převedl do typu int. Převod jsem dělal tak, že jsem hodnotu „char c a d“ převedl na hodnotu „int f a g“ a odečetl 48, na tuto hodnotu jsem přišel z ascii tabulky. Nyní jsem měl data, uměl jsem je číst, ale jak je efektivně použít? Využil jsem cyklus, kde jsem porovnával hodnoty z tabulky a tím jsem zjistil, jaké zařízení budu

```
// ctení dat
byte temperature = 0;
byte humidity = 0;
byte data[40] = {0};
if (simple_dht11_read(pinDHT11, &temperature, &humidity, data))
Serial.print("Read DHT11 failed");
return;
}

Serial.print("Sample RAW Bits: ");
simple_dht11_serial_print(data);

Serial.print("Sample OK: ");
Serial.print((int)temperature); Serial.print(" *C, ");
Serial.print((int)humidity); Serial.println(" %");

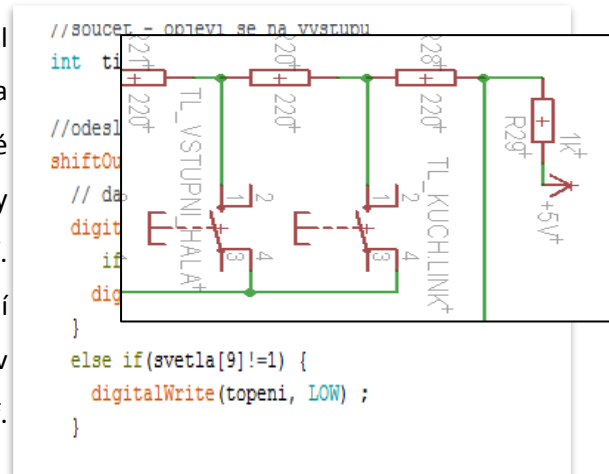
// DHT11 frekvence = 10Hz
delay(100);
```

Obrázek 18: Čtení a vypsání dat čidla DHT 11

ovládat. Další číslo vyjadřovalo, jestli chci zapnout, nebo vypnout zařízení tzn. „0“ (48) nebo „1“ (49). Tímto je vyřešené řízení z webové aplikace. Jako další schopnost inteligentního domu je zobrazování teploty a vlhkosti a to čidlem DHT11. Pro správnou funkčnost a jednodušší programování je potřeba pracovat s knihovnou určenou pro DHT (SimpleDHT.h). V programu se nastavuje komunikace DHT11 s Arduinem přes digitální pin 6. Po převedení z datového na číselný typ můžeme hodnoty vypsát například na sériovém monitoru, nebo na serveru. Nyní mám kompletně vyřešenou webovou část, ale je jasné, že ovládat dům pouze přes chytrý telefon, tablet nebo pc je celkem nevyhovující, tak jsem musel do programu zařadit i manuální ovládání. Nejjednodušším řešením by bylo připojit vypínače a každý jednotlivě připojit do pinů, ale devět pinů na arduinu je příliš mnoho, proto jsem musel řešit alternativu a použít sériovou klávesnici z tlačítek. Tato klávesnice musí být připojena do analogového vstupu, protože při stisku jednotlivého tlačítka se zmenší hodnota o určitou velikost. Čtení těchto dat je jednoduché, ale největší otázkou bylo, jak rozpoznám, co chci ovládat? Poznám to díky hodnotě, kterou arduino přečte (viz tabulka).

Zařízení	Hodnota na A0	Tlačítko A0	Hodnota na A1	Tlačítko A1
LED Půda	682	TI Půda	692	TI 9
LED Dětský pokoj	584	TI Dětský pokoj	665	TI 8
LED Ložnice	622	TI Ložnice	634	TI 7
LED Vstupní hala	313	TI Vstupní hala	597	TI 6
LED Kuchyň 1	185	TI Kuchyň 1	551	TI 5
LED Kuchyň 2	481	TI Kuchyň 2	495	TI 4
LED Obývací pokoj	538	TI Obývací pok.	424	TI 3
LED Obývací pokoj 2	655	TI Obýv. pok. 2	330	TI 2
Topení	408	TI Topení	202	TI 1

Tyto hodnoty jsem si pomocí programu vypisoval po sériové lince a poté s určitou odchylkou vypsal do tabulek (tabulka s minimální hodnotou a s maximální hodnotou). Tyto data jsou poté porovnávány v cyklu. Jakmile jsou shodná s daty v tabulkách, tak se mi vypne, nebo zapne zařízení. V programu jsem musel vyřešit rozpoznání předchozího stavu a po stisknutí tlačítka stav změnit. Vyřešil jsem to pomocí podmínky `if`.



Obrázek 19: Ovládání výstupů arduina

Ovládání tlačítka jsem použil pro ovládání v místnosti a pro ovládání z řídicí místnosti. V řídicí místnosti jsou tlačítka vedle sebe a ovládají zařízení podle přiloženého obrázku. Vyřešené vstupy mám a další problematikou budou výstupy. Ovládám devět zařízení a každé zapojit do samotného pinu je sice jednoduché, ale zbytečné. Využil jsem zde posuvný registr 74HC595. Díky kterému ovládám osm zařízení pomocí tří pinů. Jednotlivé výstupy ovládám pomocí zobrazení binárního čísla od 0 do 255. K tomuto číslu se dostanu pomocí násobení tabulky s daty zapnuto/vypnuto (1/0) a tabulky, kde mám mocniny čísla 2 (1, 2, 4, 8, 16, 32, 64, 128). Tyto násobky se následně sečtou a posuvný registr přepne piny tak, aby se objevilo na výstupu výsledné číslo v binární soustavě, tzn. binární číslo např. 164 v binární soustavě 10100100 a piny, kde je „1“ budou propouštět napětí, takže zapnuté zařízení bude na pinu u posuvného registru 7, 5 a 2. Deváté zařízení (tj. topení) jsem musel dát na samotný pin, takže ovládání je triviální. Buď zapíšu vysokou napěťovou úroveň, nebo nízkou napěťovou úroveň.

2.6 Napojení na výkonové prvky a spínací součástky

Napojení jsem realizoval pomocí plošného spoje a následným připájením součástek. K výkonovým prvkům je přivedena pouze jedna vodivá cesta, protože vnitřní strana domu je udělána jako uzemnění (GND). Jako výkonové prvky jsem zvolil LED diody, kvůli jejich spotřebě a možnosti ovládat napětím 5 V. Jako ovládací prvky jsem zvolil tlačítkovou klávesnici, abych ušetřil piny na arduinu.



Obrázek 20: Připájená LED dioda

2.6.1 Tlačítková klávesnice

Pro ušetření vstupů jsem použil klávesnici, kterou tvoří odporový dělič. Hodnotu napětí ovlivňuje, kolikátý rezistor je přiveden pomocí tlačítka na zem (GND). Na vstupu arduina se zobrazí

taková hodnota, která se rovná $U_1 = U * \frac{R1}{R_1 + n * R_2}$, kde n je počet odporů v sérii (kolikátý odpor je

uzemněný přes tlačítko), R1 = 1 kΩ a R2 = 220 Ω. Klávesnice je přivedena do Analogového vstupu na arduinu, kde se provádí digitalizace a zobrazení hodnoty od 0 do 1023. Nevýhodou této klávesnice je složitost čtení, která je ale zanedbatelná oproti ušetřenému počtu vstupů. V mém systému jsem použil tuto klávesnici pro ovládání z dané místnosti a z řídicí místnosti.

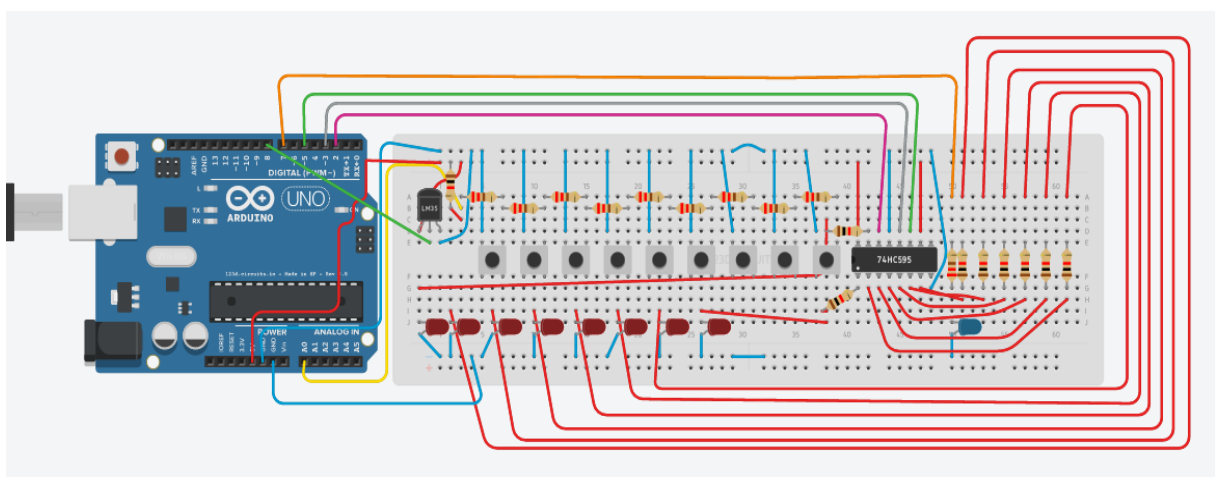
Obrázek 21: Část schéma tlačítkové klávesnice

2.6.2 LED diody připojené na posuvný registr

Zapojení LED diod jsem realizoval hlavně přes posuvný registr 74HC595, který ovládám třemi piny na arduinu. LED diody, který tento registr ovládá, znázorňují osvětlení domu. LED dioda, která je přímo připojena na pin arduina znázorňuje topení. Před každou LED diodou je připojen ochranný rezistor o hodnotě odporu 220 Ω.

2.7 Testování systému

První pokusy proběhly v online softwaru circuits.io. Výhoda používání tohoto programu je v tom, že lze zapojení otestovat, aniž by se zničili součástky. Tento online program je podobný programu fritzing. Po ověření správnosti zapojení jsem vše zapojil na desce plošného spoje. Po vyzkoušení a doladění chyb jak v programu, tak v hardwaru jsem mohl začít s tvorbou plošného spoje. Po vytvoření plošného spoje jsem zapojení otestoval zapojením arduina a nahráním programu, který ovládal dům. Během testování jsem narazil na menší problém. Tento problém se týkal rozdílných hodnot při stisknutí tlačítek. Problém jsem odstranil tím, že jsem opět nahrál program pro čtení jednotlivých hodnot při stisknutí tlačítka a zapsal do tabulky. Po odstranění této chyby bylo vše v pořádku.



Obrázek 22: Testovací zapojení v programu circuits.io

2.8 Možné vylepšení modelu

Tato verze je rozšiřitelná a lze na ni doplnit mnoho vylepšení, například žaluzie, el. zásuvky a mnoho dalších. Moje vize spočívá v tom, že bych chtěl v každé místnosti jedno arduino, které by bylo řízeno centrálním. Možnosti tohoto zhotovení by byli mnohem rozšířenější a podobali by se instalaci například od firmy Loxone. Arduina by mohli komunikovat například přes bluetooth nebo wi-fi rozhraní. Příkazy by přicházeli s arduino serveru a mohli by mít podobu mého ovládání z webové aplikace (tzn. pomocí řady čísel). Další vylepšení bych chtěl do budoucna vytvořit automatické rozsvěcování a zhasínání světel. Toto osvětlení by fungovalo propojením snímače světla a pomocí např. infračervených diod. Při určitém osvětlení by se spustilo snímání u vchodů do místnosti, snímali by se dvě místa vzdálená od sebe přibližně 20 cm a to pro identifikaci příchodu, nebo odchodu. Jakmile by v místnosti nikdo nebyl, na sčítači by byla nula, takže světlo by nesvítilo, v opačném případě, kdy by byl počet větší než nula, by světlo svítilo. Ovšem by zde bylo možné si navolit čas, kdy by toto zařízení bylo v nečinnosti

Závěr

Úkolem této maturitní práce bylo navrhnout a vytvořit model inteligentního domu. Jako řídicí jednotku jsem vybral arduino UNO. Jako první krok jsem vytvořil program, který mi umožňoval ovládat osm LED diod najednou. Ovládání těchto LED jsem vyřešil přes posuvný registr 74HC595, který se ovládá pouze třemi piny na arduinu. Tento program byl vážně jednoduchý a obsahoval mnoho nedostatků. Jako hlavní nedostatek bylo to, že při dlouhotrvajícím stisknutí tlačítka LED dioda blikala. Toto blikání způsobilo cyklení programu a při každém cyklu se změnila hodnota z log „1“ na log „0“ a naopak. Toto blikání jsem odstranil pomocí proměnné, která programu říká, či tlačítko bylo puštěno, nebo je stále smáčknuto. Po úplném vypilování tohoto programu jsem chtěl přidat ovládání z domácí sítě, ale arduino tuto funkci samostatně nezvládne, proto jsem musel použít ethernetový štít. Zde nastala další komplikace. Jelikož se nejednalo o originální štít, tak byl problém nastavit vhodnou MAC adresu štítu a celkovou komunikaci s arduinem. Po několika pokusech se mi povedlo nastavit arduino tak, aby komunikovalo s domovskou sítí bez problémů. Zároveň jsem nastavil router na statickou IP, abych nemusel po každém odpojení zjišťovat a přepisovat IP adresu v programu. Největší komplikace byla komunikace s webovým rozhraním. Po několika konzultacích s panem Bc. Janem Turoňem jsem došel k optimálnímu řešení. Toto řešení využívá HTTP protokolu, kde se zapisuje požadovaná hodnota do adresové cesty. Po vyladění na jednoduchém programu bylo na řadě vymyslet, jak odeslat data z webového prohlížeče do arduina, aby s nimi mohlo arduino pracovat. Po několika úvahách jsem se rozhodl využít dvou dekadických čísel, které znázorňují ovládaný prvek a druhé jeho stav. Zvolil jsem takovouto variantu, aby byl program rozšířitelný a to například přidáním dalšího dekadického čísla. Čísla zapsaná v adresové cestě jsou datového typu „char“ (znaky ascii tabulky), abych mohl pohodlně pracovat s těmito daty v arduinu musel jsem je převést do typu „int“ (celá čísla). Z ascii tabulky jsem se dozvěděl, že „char 0“ je roven „int 48“ a další čísla už jdou po sobě do 57 („char 9“). Tuto vědomost jsem využil v programu, kde jsem četl ascii znaky a převáděl na celá čísla, která mi symbolizovala dané zařízení a jeho stav. Po vyzkoušení se vyskytl další problém, obdobný manuálnímu ovládání. Tato chyba spočívala v tom, že příkaz uvedený ve webové aplikaci byl stále aktivní a proto mi tyto data zůstávala uležená. U prvku, kterému jsem změnil stav z webové aplikace a chtěl změnit jeho stav manuálně mi pouze požadovaný stav probliknul a vrátil se na stav, který byl odeslán z webové aplikace. Tuto chybu jsem vyřešil opět proměnnou, která říká programu, jestli se změnil požadavek z webu, nebo ne. Manuální ovládání jsem vyřešil přes analogovou klávesnici, kterou tvoří odporový dělič napětí. Toto napětí je různé podle toho, přes které tlačítko se obvod uzavře. Tuto hodnotu lze vypočítat, ale pro přesné nastavení jsem zvolil tu možnost, že aktuální hodnotu napětí jsem četl na sériové lince z arduina. Tyto hodnoty jsem zapsal do tabulky. Tlačítka ale nereagovali vždy a tak jsem byl nucen vytvořit další tabulku a

zadat hodnoty, kolem mnou naměřené, abych zabránil případnému výkyvu a tudíž i nefunkčnosti tlačítka. Celý program jsem testoval na kontaktním nepájivém poli. Po optimalizaci programu jsem vytvořil návrh plošného spoje, který symbolizuje elektroinstalaci modelu domu. Po vyleptání a doladění vodivých cest jsem desku osadil a vyzkoušel.

Celý model funguje dle požadavků a splnil jsem všechny body v zadání, proto tuto práci považuji za úspěšnou a hlavně poučnou a motivační pro můj další vývoj s tímto mikroprocesorem.



Obrázek 23: Výsledný model domu

Zdroje

DHT11: <http://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>

Program circuits.io: https://circuits.io/circuits/3428361-arduino_int_dum

Vývojový diagram: <https://www.draw.io>

Posuvný registr 74HC595: <http://arduino8.webnode.cz/news/lekce-12-posuvny-registr-74hc595/>

Lexone: <https://www.loxone.com/cscz/produkty/prehled/>

iNELS: <http://www.inels.cz/chytre-sady>

Inteligentní dům: <http://www.insighthome.eu/inteligentni-dum.html>

Eagle: <http://www.eagle.cz/info.htm>

Novinky: <http://technet.idnes.cz/online.aspx?online=1006281&strana=?>

Arduino UNO, Ethernet štít: VODA Zbyšek, Průvodce světem Arduina, 1. vydání - prosinec 2015 - ISBN 978-80-87106-90-7

Velkou část informací jsem získal v předmětech praktická cvičení, mikroprocesorová technika, elektrotechnika, informační a telekomunikační technologie a automatizační systémy.

Přílohy:

1. Vložení knihoven, definice proměnných

```
//vloženi knihoven
#include <Ethernet.h>
#include <SPI.h>
#include <SimpleDHT.h>
//pole tlacitek
int tlacitka[9] ;
//analogovy vstup tlacitek
int tlacitko = A0 ;
int tlacitko_2 = A1 ;
bool zmacknuti ;
int topeni = 7 ;
//hodnoty, které tlacitka nabývají
int hodnota_a[10]= {0,685, 660, 630, 590, 545, 490, 415, 325, 210} ;
int hodnota_b[10]= {0,700, 670, 640, 605, 560, 505, 435, 335, 195} ;

int hodnota2_a[10]= {0,675, 575,615, 305, 175, 475, 530, 650, 400} ;
int hodnota2_b[10]= {0,690, 590,625, 320, 190, 485, 545, 660, 415} ;
//pole svetel on/off
int svetla[11] ;
// pin DHT 11 - cidlo
int pinDHT11 = 6;
// promenne pro cteni z webu
char c ;
char d ;
int f ;
int g ;
```

2. Nastavení vstupů/výstupů

```
void setup() {
  //nastaveni pinu
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(tlacitko, INPUT_PULLUP) ;
  pinMode(tlacitko_2, INPUT_PULLUP) ;

  pinMode(topeni, OUTPUT) ;
  // server
  Ethernet.begin(mac, ip);
  server.begin();
  // seriova komunikace
  Serial.begin(115200) ;
}
```

3. Program, který se cyklí (hlavní program)

```
void loop(){

// DHT zacina pracovat

Serial.println(".....");

// cteni dat
byte temperature = 0;
byte humidity = 0;
byte data[40] = {0};
if (simple_dht11_read(pinDHT11, &temperature, &humidity, data)) {
Serial.print("Read DHT11 failed");
return;
}

Serial.print("Sample RAW Bits: ");
simple_dht11_serial_print(data);

Serial.print("Sample OK: ");
Serial.print((int)temperature); Serial.print(" *C, ");
Serial.print((int)humidity); Serial.println(" %");

// DHT11 frekvence = 10Hz
delay(100);

// nacteni klienta
EthernetClient client = server.available();

if(client){
//webova cast -> zobrazeni html kodu
boolean prazdnyRadek = true;
boolean hlavickaPoslana = false;
// pokud server funguje a je pripojeny, odesle data
while(client.connected() && client.available()){
// <head>
if(!hlavickaPoslana){
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");

client.println();
hlavickaPoslana = true;
// <body>
client.println("<html>");
client.println("<head><meta charset='UTF-8'></head> ");
client.println("<body>");
client.println("<a href='/?l=00'>Zjistit stav domu</a><br>");
client.println("Puda");
if(svetla[1]){
client.println("<a href='/?l=10'>VYPNI</a><br>");
}
else {
client.println("<a href='/?l=11'>ZAPNI</a><br>");
}
}
}
}
```

```

client.println("<a href= '/?l=11' >ZAPNI</a><br>");
}

client.println("Detsky pokoj");
if(svetla[2]) {
client.println("<a href= '/?l=20'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=21'>ZAPNI</a><br>");
}

client.println("Loznice");
if(svetla[3]) {
client.println("<a href= '/?l=30'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=31'>ZAPNI</a><br>");
}

client.println("Vstupni hala");
if(svetla[4]) {
client.println("<a href= '/?l=40'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=41'>ZAPNI</a><br>");
}
client.println("Kuchynska linka");
if(svetla[5]) {

client.println("<a href= '/?l=50'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=51'>ZAPNI</a><br>");
}
client.println("Kuchyn");
if(svetla[6]) {
client.println("<a href= '/?l=60'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=61'>ZAPNI</a><br>");
}
client.println("Obyvak 1");
if(svetla[7]) {
client.println("<a href= '/?l=70'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=71'>ZAPNI</a><br>");
}
client.println("Obyvakk 2 ");
if(svetla[8]) {
client.println("<a href= '/?l=80'>VYPNI<br></a>");
}
else {
client.println("<a href= '/?l=81'>ZAPNI</a><br>");
}
}

```

```

client.println("<br />");

client.println("<a href='/?l=01'>Vypnout vsechny svetla</a><br>");
client.println("Teplota je: ");
client.println((int)temperature);
client.println("*C <br>");
        if(svetla[9]) {
client.println("<a href='/?l=90'>VYPNI Topeni<br></a>");
}
else {
client.println("<a href='/?l=91'>ZAPNI topeni</a><br>");
}
client.println("Vlhkost je: ");
client.println((int)humidity);
client.println("<br>");
client.println("</body>");
client.println("</html>");
}
// cteni dat
c = client.read();
//pokud c = nicemu, tak se ukonci cteni
if(zacatekCteni && c == ' '){
    zacatekCteni = false;
}
// jakmile c = nejaky znak, zacneme cist
if(c == '?'){

}
// jakmile c = nejaky znak, zacneme cist
if(c == '?'){
    zacatekCteni = true;
}
//pokud je zadan nejaky znak, zacneme cist
if(zacatekCteni){
    if(c == 'l'){

        // cteni, protoze ctu 2 znaky a je zde i " = " tak musim jeden znak preskocit
        // c = urceni, co budu ovladat
        // d = zapnuto/vypnuto

        client.read();
        c = client.read();
        d = client.read();

/ pokud mam data, muzu vyskocit z cyklu
        break;
    }
}
}

```

```

        delay(1);
//zastavim klienta
    client.stop();

    Ethernet.maintain();
}

//kontrolni hodnoty pro cteni z webu
int f_kontrolni = f ;
int g_kontrolni = g ;

//prevedu prectene hodnoty ASCII na cisla (int)

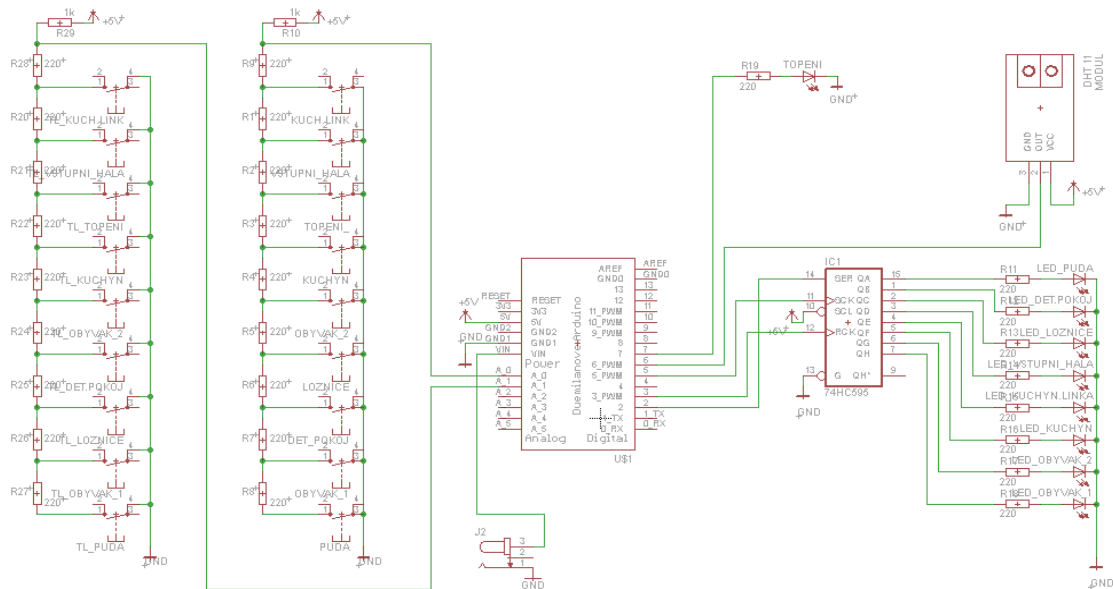
f = c ;
g = d ;
if (f_kontrolni != f || g_kontrolni != g)
{
//rozhodnovani, co se stane s daty z webu
for(int kolo = 1; kolo<11; kolo++) {
    if (f == net[kolo])
    {

        if (g==48)
        {
            svetla[kolo] = 0 ;
        }
        else if (g==49)
        {
            svetla[kolo] = 1 ;
        }
    }
}}
//cteni hodnoty v A0 -> jake tlacitko je stiskle
analogRead(tlacitko) ;
analogRead(tlacitko_2) ;
if(tlacitko != 0 || tlacitko != 1023 && tlacitko_2 != 0 || tlacitko_2 != 1023)
{
    //ochrana pred nechtenim zakmitem
    delay (1) ;
//cteni hodnoty v A0 -> jake tlacitko je stiskle
    int stav = analogRead(tlacitko) ;
    int stav_2 = analogRead(tlacitko_2) ;
    if (stav > 750 && stav_2 > 750){
        zmacknuti = false ;
    }
for (int i = 1; i<=11; i++)
{
    if(stav > hodnota_a[i] && stav < hodnota_b[i] || stav_2 > hodnota2_a[i] && stav_2 < hodnota2_b[i])
    {

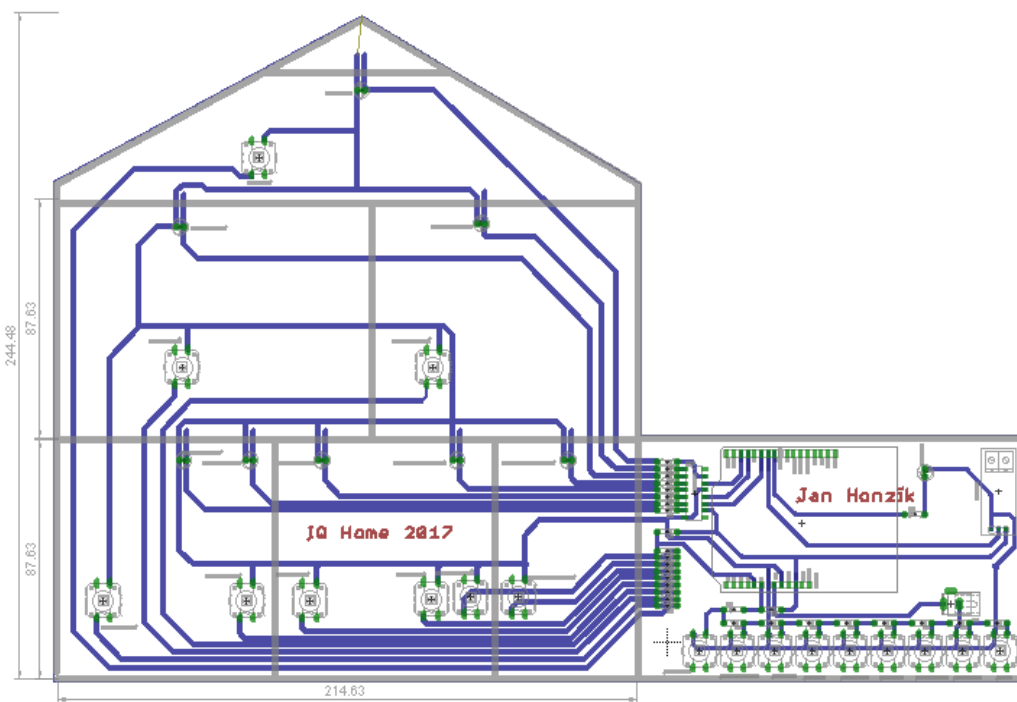
        if (svetla[i] != 0 )
        {
            svetla[i] = 0 ;
            zmacknuti = true ;
        }
        else if(svetla[i] == 0) {
            svetla[i] = 1 ;
            zmacknuti = true ;
        }
    }
}
}} }

```

4. Schéma modelu inteligentního domu



5. Návrh plošného spoje modelu inteligentního domu



6. Návrh plošného spoje – vodivé cesty a díry

