

GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Zařízení pro realizaci chytré domácnosti

Maturitní práce

Autor: Vladislav Aulich

Třída: 4.C

Školní rok: 2020/2021

Předmět: Informatika

Vedoucí práce: Emil Miler

Praha, 2021



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Vladislav Aulich

Třída: 4.C

Školní rok: 2020/2021

Platnost zadání: 30. 9. 2021

Vedoucí práce: Emil Miler

Název práce: Zařízení pro realizaci chytré domácnosti

Pokyny pro vypracování:

Cílem práce je vytvořit chytrou domácnost, konkrétně ovládání světel pomocí mikrokontroleru. Systém bude možné ovládat pomocí jednoduchého rozhraní z počítače či mobilního zařízení. Systém bude libovolně rozšiřitelný o další čidla a senzory.

URL repozitáře:

https://github.com/aulisek/chytra_domacnost

vedoucí práce

student

V Praze dne 30. 10. 2020

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 8. dubna 2021

Vladislav Aulich

Poděkování

Touto cestou bych rád poděkoval svému vedoucímu práce Emilu Milerovi za zapůjčení hardwarového vybavení, podporu a odbornou pomoc při tvorbě projektu. Dále bych chěl poděkovat svému bratrovi za pomoc při stavbě 3D tiskárny a odstraňováním následných vad tisku a své sestře za pomoc při testování a motivaci k dokončení projektu.

Abstrakt

Tento projekt se zabývá realizací chytré domácnosti. Pro tvorbu projektu byl zvolen mikrokontrolér Arduino uno a čip ESP32. Komunikace mezi zařízeními je realizovaná bezdrátově za použití otevřené rádiové frekvence 433 MHz. V rámci projektu bylo vytvořeno jednoduché webové rozhraní v HTML. Součástí projektu je i výroba hardware s použitím metody 3D tisku na tiskárně Ender 3 pro. Vývoj kódu probíhal v prostředí Arduino IDE, pro trasování změn byl využit verzovací systém git.

Klíčová slova

Chytrá domácnost, automatizace, ESP32, Arduino, Arduino IDE, 433 MHz, HTML

Abstract

This project deals with the implementation of a smart home. The project is created using an Arduino uno microcontroller and an ESP32. Communication between devices is realized wirelessly using an open radio frequency of 433 MHz. Within the project, a simple web interface in HTML was created. The project also includes the production of hardware using the 3D printing method on the Ender 3 pro printer. The code was developed in the Arduino IDE environment, and the git versioning system was used to trace the changes.

Keywords

smart home, automation, ESP32, Arduino, Arduino IDE, 433 MHz, HTML

Obsah

1	Teoretická část	3
1.1	Úvod	3
1.2	Cíl práce	3
2	Implementace	5
2.1	Schéma řešení	5
2.1.1	Centrála	5
2.1.2	Koncové zařízení	7
2.2	Návrh prototypu	8
2.2.1	Centrála	8
2.2.2	Koncové zařízení	9
2.3	Výroba prototypu	10
2.3.1	3D tisk	10
2.3.2	Kalkulace nákladů	10
2.4	Možnosti automatizace	12
2.5	Návrhy na vylepšení	13
2.5.1	Uživatelské rozhraní	13
2.5.2	Zařízení	13
2.6	Testování	13
3	Technická dokumentace	15
	Závěr	19
	Seznam použité literatury	21
	Seznam obrázků	23
	Seznam tabulek	24

1. Teoretická část

1.1 Úvod

Toto téma jsem si zvolil, protože jsem chtěl blíže prozkoumat práci s platformou Arduino a ESP32. S programováním těchto zařízení jsem měl minimální zkušenosti, proto pro mě byla práce na projektu výzvou k objevování nového. Použití komunikace na rádiové frekvenci jsem zvolil z důvodu široké škály použití a velkého množství příkladů. Zároveň jsem měl doma nevyužívaný ovladač pracující s touto frekvencí.

Motivací k výběru tématu „chytré domácnosti“ mi bylo její čím dál větší nasazování v domácnostech a snaha vytvořit si ji po svém. Na mnohých komerčních řešeních mi totiž nevyhovoval způsob ovládání, nutnost dokupovat různé sety, stejně jako velký zásah do soukromí uživatelů.

1.2 Cíl práce

Cílem této práce je vytvořit zařízení pro realizaci chytré domácnosti. Zařízení si klade za cíl ovládat spotřebiče uživatele tzv. „na dálku“. Součástí řešení musí být uživatelské rozhraní, možnost další automatizace a možnost ovládání „offline“.

2. Implementace

2.1 Schéma řešení

Pro dosažení cíle práce jsem projekt rozdělil na dvě nezávislá zařízení:

- Centrály - brány ovládající další zařízení
- Koncového zařízení ovládajícího spotřebič

Pro každé zařízení jsem pak implementoval potřebné funkce. Dále bylo potřeba zajistit jejich vzájemnou komunikaci.

2.1.1 Centrála

Pro „centrálu“ jsem si vybral platformu ESP32 (konkrétně model NODE-MCU32-S¹) a to hned z několika důvodů. Čip má integrovanou wifi, k dispozici je velké množství dokumentace, hardwaru s příklady a knihovny. Další výhodou je velká komunita, což může pomoci při řešení problémů. Čip lze také integrovat do prostředí Arduino IDE, což umožňuje snadnější práci při tvorbě kódu.

Při výběru jsem zvažoval i platformu raspberry pi, ale odradila mě přítomnost operačního systému, který je zbytečný pro tak malý projekt a velké pořizovací náklady oproti ESP32.

Centrála zajišťuje několik funkcí:

- Komunikace s uživatelem
- Komunikace mezi zařízeními
- Správa uložených zařízení

Komunikace s uživatelem

Pro komunikaci s uživatelem je vytvořeno jednoduché webové rozhraní umožňující dynamicky vypsat uložená zařízení a přidat nové. Pro tvorbu webového rozhraní jsem zvolil knihovnu ESPAsyncWebServer². Tato knihovna má dobře zpracovanou dokumentaci a pro nasazení v projektu se hodí svými funkcemi. Samotné připojení čipu k Wi-Fi je realizováno pomocí knihovny WiFi³.

První verze kódu měly implementovaný jednoduchý synchronní webserver pomocí knihovny WiFi, obsažené v základní verzi Arduina IDE. Toto řešení nebylo vhodné, protože numožňovalo připojení více uživatelů k rozhraní naráz.

Další implementovanou funkcí byla autentifikace uživatele. Tato funkce je zajištěná při odesílání requestu pomocí knihovny ESPAsyncWebServer.

¹<https://rpishop.cz/iot/2605-waveshare-nodemcu-32s-esp32-wifi-bluetooth-vyvojova-deska.html>

²<https://github.com/me-no-dev/ESPAsyncWebServer>

³<https://github.com/arduino-libraries/WiFi>

Pro zjednodušení přístupu k uživatelskému rozhraní je na zařízení spuštěn mDNS server na adrese `http://esp32.local`. Tato funkce je realizována pomocí knihovny ESPmDNS⁴.

Uživatelské rozhraní je vytvořeno pomocí HTML. HTML soubor je uložen ve vnitřní flash paměti čipu ESP. K ovládání filesystému SPIF je užito knihovny SPIFFS⁵. Tato knihovna obsahuje všechny potřebné funkce.

Pro přidání nového zařízení má uživatel na výběr mezi přidáním koupeného komerčního zařízení nebo přidáním zařízení vyrobeného v rámci tohoto projektu. Postup při zadávání nového zařízení je nastíněn ve webovém rozhraní.

Komunikace programu a webového rozhraní je zajištěna pomocí HTTP GET requestů, které program zpracuje a provede potřebné akce.

Komunikace mezi zařízeními

Komunikace mezi zařízeními je realizovaná bezdrátově po frekvenci 433 MHz. Tento druh jsem zvolil z důvodu rozšířenosti a nízkých pořizovacích nákladů modulů⁶. Další nespornou výhodou je dosah, který v optimálním prostředí může být až 200 m.

Další ze zvažovaných řešení byla realizace pomocí Wi-Fi nebo GSM modulu. V prvním případě jsem jako nevýhodu viděl dosah (závislost na dostupnosti připojení). Dále by nebylo tak snadné ovládat zařízení pomocí ovladače. V druhém případě byla nevýhoda cena a potřeba realizace připojení pomocí telefonního operátora.

Ovládání modulů jsem chtěl realizovat pomocí jednoduché knihovny VirtualWire, ale zjistil jsem, že není funkční na čipech ESP. K ovládání jsem tak použil knihovnu RadioHead.

Pro ovládání „Konečného zařízení“ je vyslána zpráva obsahující ID zařízení a tag ON nebo OFF, „konečné zařízení“ následně odchyti ID a provede příkaz. Tento systém je do budoucna rozšiřitelný o další tagy pro zařízení, které potřebují k ovládní více příkazů než ON nebo OFF.

Tento navržený systém jsem implementoval, nicméně jsem narazil na problém s ovládáním pomocí komerčního ovladače. K tomuto účelu jsem začlenil knihovnu RCswitch⁷, která umožňuje zobrazit protokol, na jehož základě právě komerční zařízení komunikují. Poté dokáže vysílat tímto protokolem.

Ve svém řešení jsem chtěl použít knihovny obě a rozdělit tak ovládání na dvě části, podle toho, jakým způsobem mají komunikovat. Při realizaci tohoto jsem však narazil na problém, že není možné bezdrátový vysílač ovládat dvěma knihovnami současně.

Nakonec jsem vybral knihovnu RCswitch z důvodu větší spolehlivosti při přenosu vysílání a možného rozšíření na ovládání komerčních zařízení (nebo ovládání komerčními zařízeními). Toto řešení sebou nese nutnost dvou kódů, pro zapnutí a vypnutí (při použití již výše zmiňovaného ovladače).

⁴<https://github.com/espressif/arduino-esp32/tree/master/libraries/ESPmDNS>

⁵<https://github.com/espressif/arduino-esp32/tree/master/libraries/SPIFFS>

⁶<https://www.laskarduino.cz/sada-vysilac-prijimac-433mhz/>

⁷<https://github.com/sui77/rc-switch>

Správa uložených zařízení

Zařízení se ukládají přímo do flash paměti ESP32 o velikosti 4 MB. Soubor je nazván „zarizeni.csv“. Zařízení se ukládají ve formátu CSV (comma separate value). Pro ovládání filesystému byla zvolena výše zmíněná knihovna SPIFFS. Pro „parsování“ csv souboru jsem využil již hotové řešení za pomoci knihovny CSVparser⁸. Knihovna je pěkně zadokumentovaná a obsahuje vše, co jsem pro čtení CSV potřeboval.

Do sloupce „název“ se uloží název, který si zadá sám uživatel v uživatelském rozhraní. Do sloupce „kod_ovladac“ se v případě zařízení vyrobeného v rámci projektu uloží „x“, protože k ovládání stačí ID zařízení, které je v tomto případě ve sloupci „kod_zarizeni“.

Při zadávání komerčního zařízení záleží na způsobu ovládání daného zařízení. Při tvorbě jsem vycházel z mého ovladače, který má zvlášť kód pro vypnutí a zapnutí. Jiná zařízení mají odlišná schémata.

Pro implementaci takového typu zařízení je za potřebí analyzovat, jaký způsob ovládání zařízení používá. K tomu slouží example kód knihovny RCswitch. Pro funkčnost ovládání je také potřeba prostudovat dokumentaci k této knihovně a dopsat správný typ ovládání do místa v kódu (Toto místo je označeno přímo v kódu).

2.1.2 Koncové zařízení

Toto je hlavní částí mého projektu. Koncové zařízení provádí následující funkce:

- přijímá požadavek od centrály
- zapne/vypne spotřebič
- reaguje na pohyb
- obsahuje senzor světla
- reaguje na vstup uživatele z tlačítka

Pro „koncová zařízení“ jsem zvolil jako vývojovou platformu Arduino uno⁹. Po odladění kódu a hardwaru jsem vyrobil prototyp, který již využíval Arduino Pro mini¹⁰ s pěti voltovou logikou.

Pro realizaci zapínání a vypínání spotřebičů jsem zvolil ovládání pomocí relé. Pro snadnější ovládání jsem vybral již hotový modul pro arduino¹¹, který pokud je na vstupu logická 1 vypnutý, pokud je přítomna zem, tak je zapnutý (je tzv. negovaný).

Reakce na pohyb je zajišťována PIR senzorem¹². Opět byl zakoupen modul, který bylo snazší implementovat. Aby nedocházelo k zapnutí světla ve dne, byl k senzoru přidán fotorezistor, který reaguje na hladinu světla v místnosti. Při tvorbě kódu pro fotorezistor jsem narazil na problém s rozeznáváním světla od tmy. Fotorezistor v mém zapojení totiž neukazoval konstantní hodnoty.

Při prvním pokusu ovládání jsem nastavil pevnou hranici pro tmu, kde jsem porovnával napětí na

⁸<https://github.com/michalmonday/CSV-Parser-for-Arduino>

⁹<https://www.laskarduino.cz/arduino-uno-r3--atmega328p--klon/>

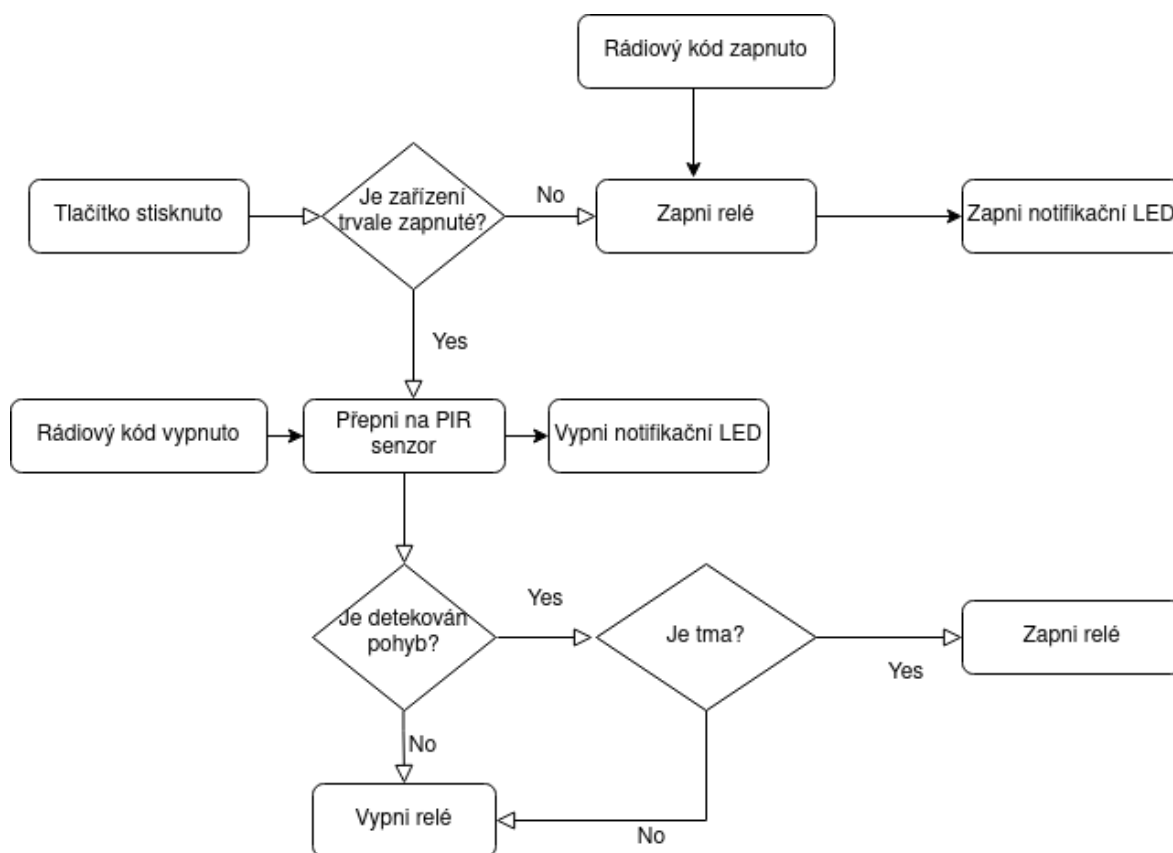
¹⁰<https://www.laskarduino.cz/arduino-pro-mini--atmega328-5v-16mhz--klon/>

¹¹<https://drotek.cz/arduino/2954-modul-rele-5v-1-kanal-opticky-oddeleno.html>

¹²<https://www.laskarduino.cz/arduino-pir-detektor-pohybu-hc-sr501/>

analogovém vstupu s pevně danou hodnotou. Toto řešení se neukázalo jako vhodné ve chvíli, kdy fotorezistor ukazoval hodnoty blízké k hranici, docházelo tak k rychlému spínání a vypínání světla. Ve finálním kódu jsem použil hystereze a k tomuto efektu nedochází. Nevýhoda tohoto řešení je, že si uživatel musí nastavit hranici tmy v kódu. Toto by se dalo vyřešit náhradou rezistoru za trimmer, kterým by uživatel měnil hodnotu odporu a s tím i napětí na analogovém vstupu.

Při hlubším hledání na internetu jsem objevil právě takový modul¹³, který umožňoval trimmerem měnit „citlivost na tmu“. Proto jsem ho zakoupil a ve výsledném řešení je implementován. Samotný modul má dva výstupy. Jeden je analogový ukazující „hladinu tmy“ a druhý je digitální, který má buď logickou jedničku na světle, zem při tmě. Toto zapojení se tak i hodí k negovanému relé, kde lze zapojit modul napřímo k ovládání relé bez nutnosti mikrokontroleru. V repozitáři jsou tak dva kódy, v zařízení je použit kód nazvaný `prijimac_modul_svetlo.ino`.



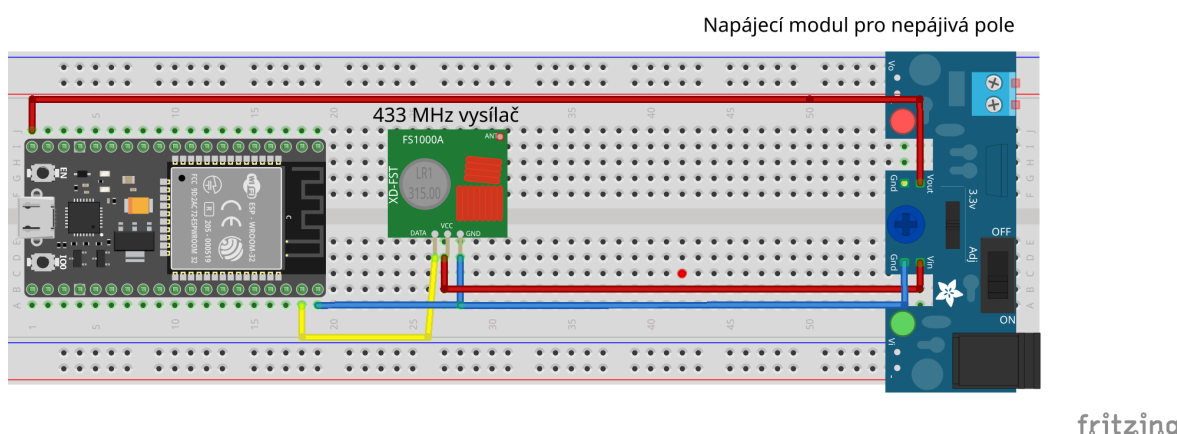
Obrázek 2.1: Schéma činností koncového zařízení

2.2 Návrh prototypu

2.2.1 Centrála

Centrálu jsem se rozhodl ponechat v nepájivém kontaktním poli a vyrobit k ní krabičku na 3D tiskárně. Návrh krabičky pro 3D tisk jsem tvořil v programu thinkercad.

¹³<https://www.laskarduino.cz/arduino-svetelny-senzor--4-pin-modul/>



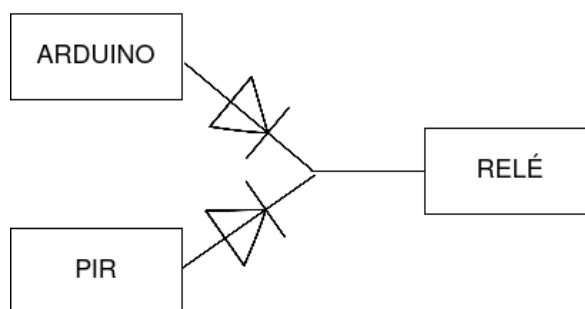
Obrázek 2.2: Schéma zapojení centrály

2.2.2 Koncové zařízení

Pokus 1

Oddělení na snímání pohybu pomocí PIR čidla a ovládání pomocí tlačítka (ovladače) jsem původně chtěl rozdělit hardwarově. Jedna část by bylo Arduino, které přijímá kód po rádiu a vstup uživatele z tlačítka a druhá část by byl PIR senzor, který obstarává vše ostatní.

Protože PIR senzor má vlastní časovač, logickou 1 dává na výstup vždy podle uživatele nadefinovanou dobu od posledního pohybu. Toho jsem chtěl využít a spojit výstup arduina pomocí dvou diod s výstupem PIR senzoru a připojit to na vstup relé. Toto řešení však sebou neslo spoustu nevýhod.



Obrázek 2.3: Schéma zapojení relé a diody

Toto řešení nefungovalo perfektně, kvůli úbytku napětí na diodě (0.7 V), takže výstup s PIR nebyl iniciován jako logická 1. Další nevýhodou byl fakt, že používám negované relé. Tyto problémy by šly vyřešit použitím tranzistoru, nicméně i kvůli detektoru hladiny světla jsem toto řešení zavrhl.

Pokus 2

Po prvním nezdařilém pokusu jsem přešel na kompletní ovládání arduinem. Nejprve jsem si všechny potřebné díly spojil v nepájivém kontaktním poli a potom jsem ověřil funkčnost kódu. Takto vyrobené zařízení splňovalo všechny požadavky, takže jsem se rozhodl zařízení přenést do kompaktnější podoby.

Opět jsem vytvořil návrh krabičky pro 3D tisk. Při návrhu krabičky pro PIR senzor jsem vycházel již z hotového modelu¹⁴ staženého z portálu thinkverse, který jsem obohatil o díru na fotorezistor a notifikační LED. Návrh je od autora hills8, publikován je pod licencí creative commons-attribution. Návrh krabičky pro zařízení jsem již vytvářel sám. Moduly jsem následně přenesl na zakoupenou PCB návrhovou desku podle schématu.

2.3 Výroba prototypu

Tisk součástek jsem prováděl na tiskárně Ender 3 pro, kterou jsem si sám složil. Při pájení součástek na desku jsem narážel na nedostatečné vybavení, to jsem nakonec vyřešil zapůjčením vybavení od vedoucího práce. Užitečné instrukce k letování jsem našel na portálu youtube ve videu „Basic Soldering Lesson 1 - "Solder Flux"“ z kanálu paceworldwide.¹⁵

2.3.1 3D tisk

Tisk probíhal na již zmíněné tiskárně Creality Ender 3 pro¹⁶. Tisknul jsem na originální magnetickou podložku, na tiskárně byl vyměněn podavač filamentu a přidán BLtouch zajišťující automatické „vyrovnaní“ desky. Materiál pro tisk jsem zvolil PLA, konkrétně jsem měl PLA od Creality¹⁷ v šedé barvě.

Parametry tisku

- Podpěry: NE
- Brim: NE
- Výška vrstvy: 0,2
- Výplň: 10% (mřížka)
- Filament: PLA
- Teplota podložka, první vrstva: 60°C
- Teplota podložka, ostatní vrstvy: 50°C
- Teplota tryska, první vrstva: 210°C
- Teplota tryska, ostatní vrstvy: 206°C

2.3.2 Kalkulace nákladů

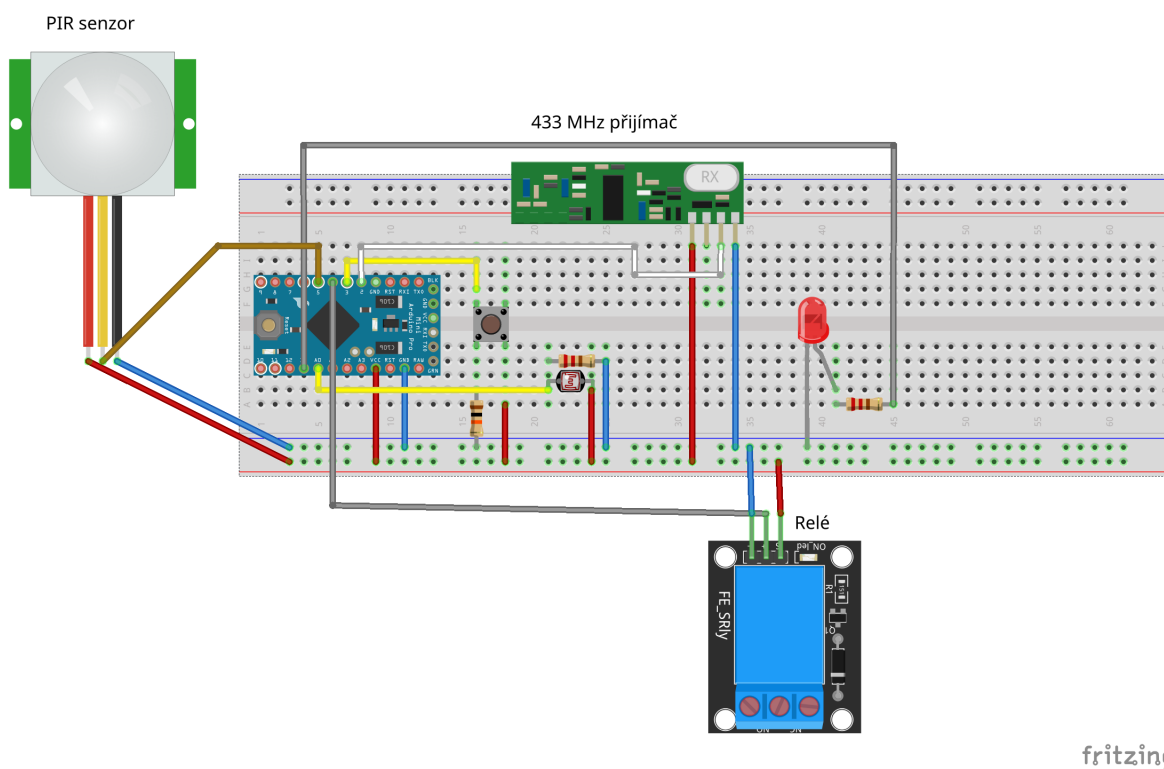
Celkové náklady na „centrálu“ bez 3D tisku jsou přibližně 300 Kč. Celkové náklady za prototyp „koncového zařízení“ bez 3D tisku je 265 Kč.

¹⁴<https://www.thingiverse.com/thing:2845890>

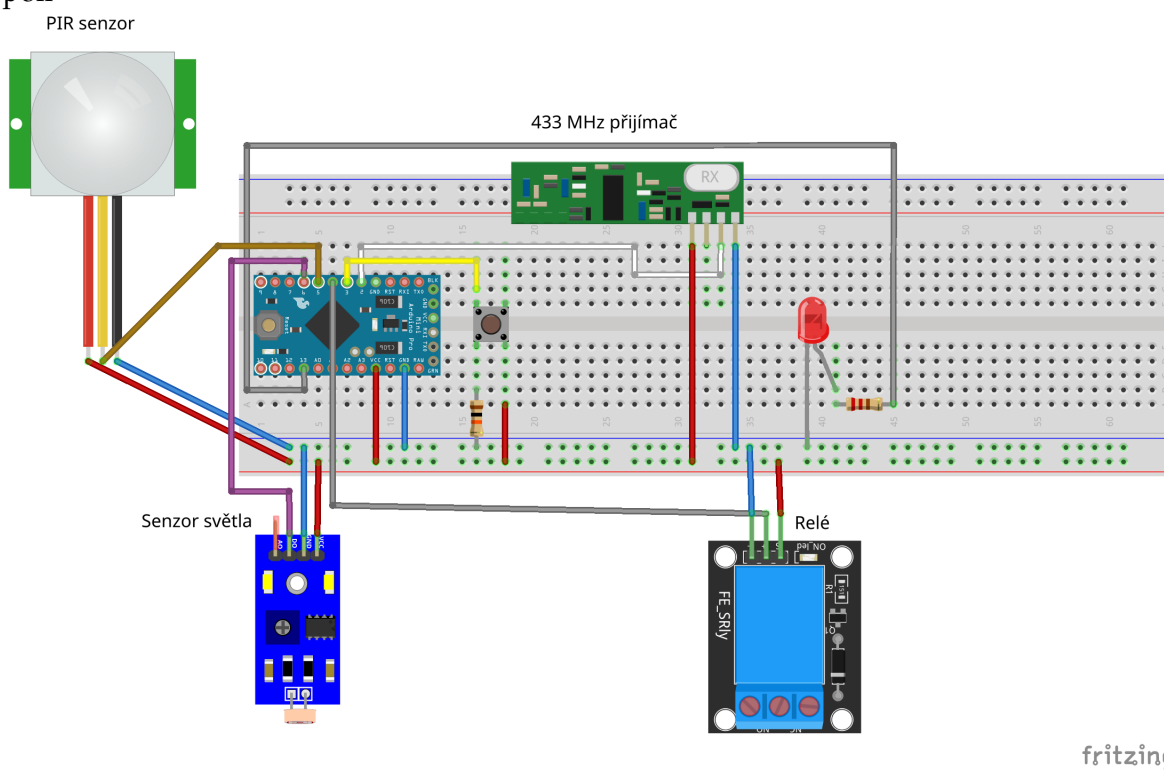
¹⁵<https://www.youtube.com/watch?v=vIT4ra6Mo0s>

¹⁶<https://www.creality3dshop.eu/collections/ender-series-3d-printer/products/creality3d-ender-3-pro-high-precision-3d-printer>

¹⁷<https://www.creality3dshop.eu/collections/filament/products/3d-printer-silver-color-pla-filament-1-75mm-ac>



Obrázek 2.4: Schéma zapojení koncového zařízení s fotorezistorem v nepájivém kontaktním poli



Obrázek 2.5: Schéma zapojení koncového zařízení se senzorem světla v nepájivém kontaktním poli

Počet kusů	Název	Cena [Kč]
1	NodeMCU-32S ESP32	249
2	433 MHz vysílač a přijímač	79
2	Spirálová anténa 433 MHz	10
1	Mikrospínač	4
2	Relé modul s optickým oddělením	65
1	Rezistor 10k	1
1	Arduino Pro Mini	98
1	Arduino uno	599
1	PCB prototypová deska	18
1	PIR detektor pohybu	38
1	Detektor tmy	20
1	Konektor USB	10
1	USB napájecí kabel	30

Tabulka 2.1: Kalkulace nákladů

2.4 Možnosti automatizace

Díky ovládání pomocí HTTP GET requestů jsou možnosti automatizace pomocí softwaru třetích stran poměrně jednoduché. K automatizaci lze využít aplikaci Tasker¹⁸ pro Android, která nabízí velice rozsáhlé možnosti. Tato aplikace je placená, podobné funkce nabízí i aplikace Automate¹⁹, která je zdarma.

Uživatel si v aplikaci může nadefinovat chování „co se stane když“ v reakci na nějaký podmět. Lze si například nadefinovat zapnutí světla pokud se připojíme k domácí wi-fi, nacházíme-li se na nějakých GPS souřadnicích, v reakci na čas, při příjmu oznámení telefonu a mnohem více. To, co tvoří chytrou domácnost opravdu chytrou je právě automatizace. Na tu se projekt nezaměřoval, nicméně implementace do projektů třetích stran není problém.

Struktura requestu pro zařízení:

```
http://<IP_adresa_esp>/get?kod=<KOD>
```

Nebo při použití mDNS

```
http://<esp32.local>/get?kod=<KOD>
```

Za kód dosadíme buď kód pro vypnutí nebo zapnutí zařízení.

¹⁸https://play.google.com/store/apps/details?id=net.dinglish.android.taskerm&hl=en_US&gl=US

¹⁹https://play.google.com/store/apps/details?id=com.llamalab.automate&hl=en_US&gl=US

2.5 Návrhy na vylepšení

2.5.1 Uživatelské rozhraní

Při zpracování rešerše na začátku projektu jsem objevil zajímavé řešení uživatelského rozhraní pomocí Blynk²⁰. Tato platforma má rozsáhlé možnosti, zejména pro „Internet věcí“. To by mé řešení mohlo rozšířit mimo domácí síť za využití cloudu. Samotný cloud je možné nainstalovat na vlastní zařízení (například raspberry pi), můžete tak mít dohled nad svými daty. Zároveň projekt nabízí vytvoření mobilní aplikace pro Android, mohl bych tak řešení povýšit o mnoho pater výše, co se týče uživatelské přívětivosti. Projekt je open-source, mobilní aplikaci je pak možné využívat i komerčně. Implementace do projektu by nebyla vůbec složitá.

2.5.2 Zařízení

Nevýhoda „koncového zařízení“ je napájení, kdy potřebujeme externí zdroj. Toto řešení jsem zvolil kvůli své jednoduchosti, zároveň by zařízení pro zapojení do sítě potřebovalo revizi podle Vyhlášky č. 50/1978 Sb. Českého úřadu bezpečnosti práce a Českého báňského úřadu o odborné způsobilosti v elektrotechnice²¹.

Jinak by zařízení bylo schopné obstarávat všechny funkce, vešlo by se do elektroinstalační krabice místo normálního vypínače. K této minimalizaci se také nabízí optimalizace použitých součástek (používat základní komponenty místo modulů, například relé) a sestavení plošného spoje. Ten by byl spolehlivější, menší a pro toto nasazení daleko vhodnější.

Dále bych chtěl do projektu začlenit ovládání naší vjezdové brány na zahradu, která pracuje s plovoucím kódem na frekvenci 433 MHz. Toto by znamenalo možnost otevírat bránu pomocí telefonu místo ovladače a další možnosti automatizace, například automatické zavlažování.

Dále jsem našel zajímavý projekt zabývající se ovládání rolet²². Tento projekt by nebyl problém realizovat a k ovládání použít své vytvořené schéma a rozhraní. Naše domácnost by se tak stále zase o trochu chytřejší.

2.6 Testování

Celé zařízení jsem otestoval, z dosažených výsledků jsem byl však zklamaný. Ačkoliv jsem bezdrátový vysílač napájel maximálním možným napětím, nedosahoval tak dobrých přenosových vzdáleností. Dle mé hypotézy pravděpodobně bude chyba na straně vysílače, zakoupením dražšího modulu s deklarovanými vzdálenostmi by se asi problém vyřešil. Žádné další problémy jsem však nezaznamenal.

²⁰<https://blynk.io/>

²¹<https://www.zakonyprolidi.cz/cs/1978-50>

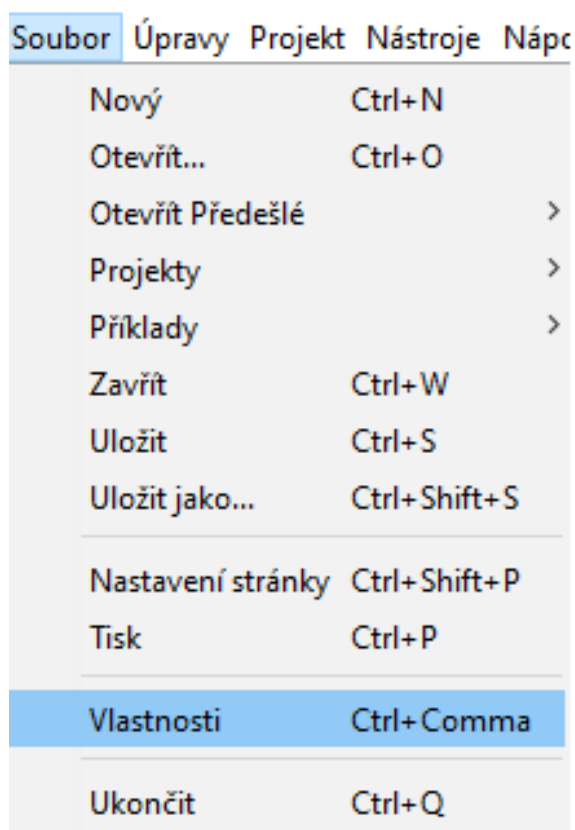
²²<https://www.instructables.com/Arduino-Motorized-Roller-Blinds/>

3. Technická dokumentace

„Koncové zařízení“ je samostatně funkčním celkem, stačí připojit pomocí přiloženého usb kabelu k napájení. Zařízení vyžaduje 5V, což je běžné napětí USB. Po připojení k napájení je zařízení v provozu a čeká na vstup uživatele. Ve výchozím stavu je relé rozepnuté, při připojování ovládaného zařízení tak stačí respektovat značení na relé. Pokud ovládáte zařízení běžné elektroinstalace, přerušujete fázi (zapojení jako u vypínače). Na pin COM tedy přiveďte napájení. Pokud vstup do zařízení připojíte k NO (normally open), bude obvod uzavřen při rozepnutém stavu relé (to znamená i při odpojení napájení k modulu). Připojením k NC (normally closed) bude obvod uzavřen při sepnutí relé. Pro potřeby projektu zapojte dle obrázku.

Při uvádění „centrály“ do provozu potřebujete zadat údaje k Wi-Fi síti (SSID a heslo). Zařízení podporuje standardy IEEE 802.11B, IEEE 802.11G, IEEE 802.11N a zabezpečení WPA/WPA2/WPA2-Enterprise, WPS. Údaje se zadávají do kódu pro zařízení, který je následně nutné zkompilevat a nahrát do čipu. Postup uvádím v prostředí Arduino IDE. Návod k instalaci je dostupný v anglickém jazyce na [randomnerdtutorials](https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/)¹.

1. Otevřete soubor `centrala.ino`² pomocí Arduino IDE.
2. Na kartě Soubor otevřete vlastnosti.

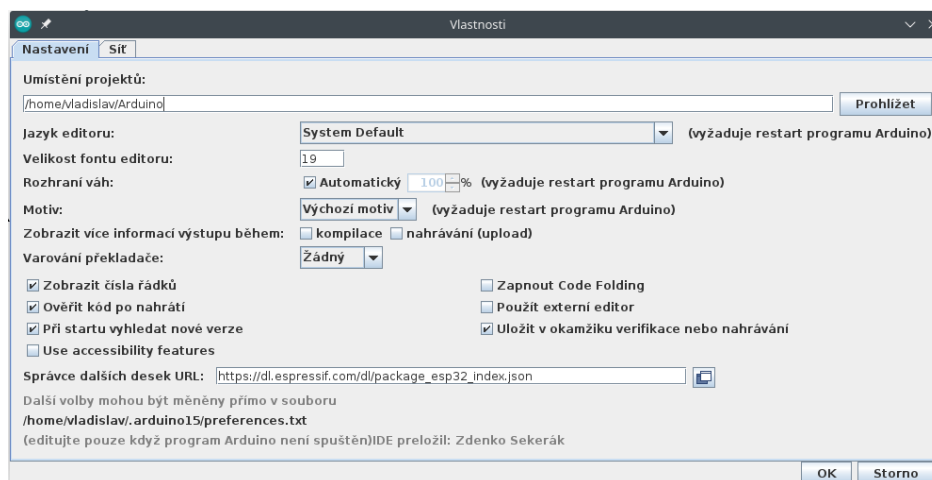


Obrázek 3.1: Přístup ke kartě vlastnosti

¹<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

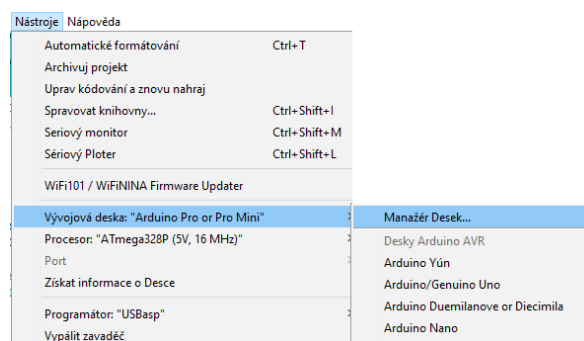
²https://github.com/aulisek/chytra_domacnost/blob/main/SRC/centrala/centrala.ino

- Do pole „Správce dalších desek URL“ vložte: https://dl.espressif.com/dl/package_esp32_index.json
- Potvrďte OK.



Obrázek 3.2: Nastavení na kartě vlastnosti

- Na kartě nástroje zvolte Manažér desek.

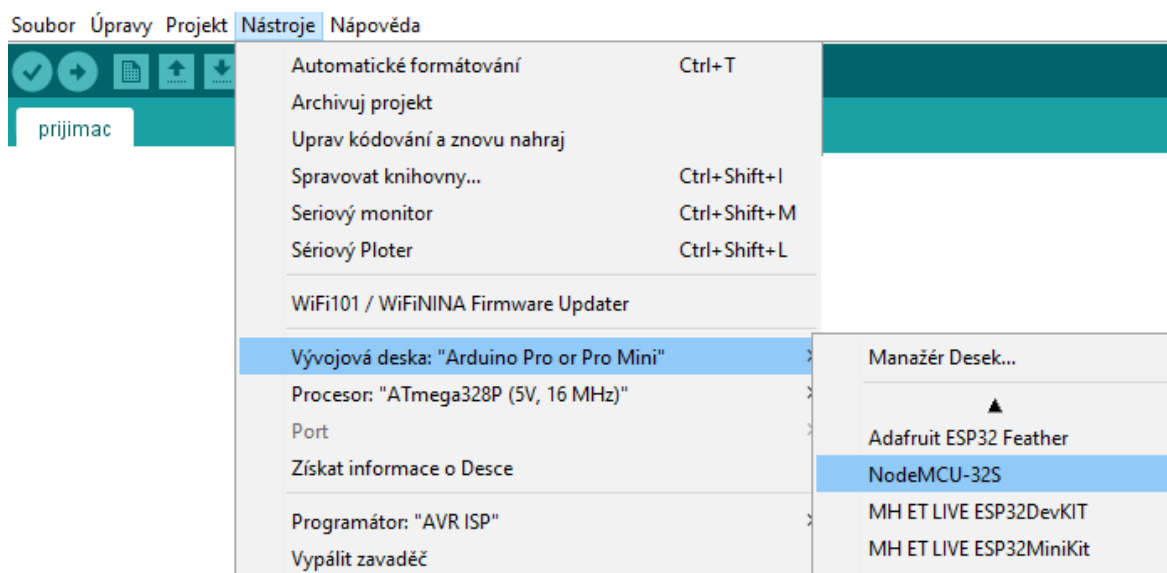


Obrázek 3.3: Přístup k Manažéru desek

- Do vyhledávání zadejte „esp“.
- Klikněte na instalovat a vyčkejte než se proces dokončí.
- Po úspěšné instalaci zvolte „Zavřít“.
- Připojte zařízení k PC pomocí micro USB kabelu.
- Na kartě nástroje vyberte port, ke kterému je zařízení připojeno.
- Na kartě „Nástroje“ vyberte zařízení NODEMCU ESP-32S
- Klikněte na šipku v levém horním rohu pro kompilaci a nahrání kódu.
- Pro ověření můžete na kartě „Nástroje“ zobrazit „Sériový monitor“, nastavte v levém dolním rohu rychlost na 115200 baudů.
- V Sériovém monitoru by se Vám měli ukázat případné chyby.
- Odpojte čip od PC a připojte napájecí adaptér.
- Připojte se k uživatelskému rozhraní na adrese <http://esp32.local>.
- Do administračního prostředí je potřeba autentifikace, výchozí jméno je admin a heslo admin, změnit lze v kódu na řádce 14 a 15.



Obrázek 3.4: Instalace desky v prostředí manažéru



Obrázek 3.5: Zvolení zařízení ze seznamu

Závěr

Závěr obsahuje shrnutí práce a vyjadřuje se k míře splnění jejího zadání. Dále by se zde mělo objevit sebehodnocení studenta a informace o tom, co nového se naučil a jak vnímal svou práci na projektu.

V rámci projektu jsem vytvořil funkční soustavu zařízení umožňující realizaci chytré domácnosti. Toto zařízení splňuje všechny body zadání. Se zařízením jsem spokojený, splňuje všechny mé požadavky. Do budoucna se nabízí rozšíření a vylepšení zařízení.

Při tvorbě projektu jsem si osvojil mnoho nových dovedností. Při softwarovém návrhu jsem se blíže seznámil s prostředím Arduino IDE, práci s knihovnamí a dokumentací. Naučil jsem se práci se senzory a způsoby jejich připojení. Při tvorbě webového prostředí nasazení HTML a kaskádových stylů. K zaznamenávání verzí projektu jsem používal verzovací systém git, změny jsem nahrával na server github. Bylo to moje první setkání s verzovacími systémy, ze začátku jsem trochu tápal, jak ho přesně použít. Při tvorbě kódu jsem však ocenil přehlednost a vracení se ke starším verzím. Systém jsem ovládal z příkazové řádky, naučil jsem se tak spravovat svůj soborový systém a efektivní práci s ním. Při tvorbě dokumentace jsem se seznámil se sázecím systémem \LaTeX , který pro mě byl nový.

Realizace hardwarové části mi přinesla zkušenost s letováním, nahráváním programu do Arduina pro mini pomocí programátoru mimo prostředí Arduino IDE. Také jsem se naučil základy programu fritzing při tvorbě schémat zapojení modulů. Naučil jsem se základy 3D modelování pomocí softwaru thinkercad, následné převedení předlohy modelu do hotové podoby použitím metody 3D tisku. S tím se pojí i samotná údržba 3D tiskárny. Po jejím složení jsem se rozhodl pro několik úprav. Seznámil jsem se s firmwarem Marlin, naučil jsem se ho upravovat, zkompileovat a nahrát ho do tiskárny za použití USBisp programátoru. Dále jsem se dozvěděl více o materiálech pro 3D tisk a potřebným nastavením parametrů tisku.

Seznam použité literatury

- [] *Arduino converting value from file.read() to const char**. URL: <https://stackoverflow.com/questions/61987375/arduino-converting-value-from-file-read-to-const-char>.
- [Mal17] Martin Malý. *Hradla, volty, jednočipy: úvod do bastlení*. Praha: CZ.NIC, 2017. ISBN: 978-80-88168-23-2.
- [Mil87] Boleslav Černoch Milan Syrovátko. *Polovodičová technika*. Praha: SNTL - Nakladatelství technické literatury, 1987.
- [Oldoo] Přemysl Šedivý Oldřich Lepil. *Fyzika pro gymnázia: elektřina a magnetismus*. Praha: Prometheus, 2000. ISBN: 80-7196-202-3.
- [San] Nuno Santos. *techtutorialsx*. URL: <https://techtutorialsx.com/>.
- [Šra79] Pavel Šrait. *Od krystalky k modelům s tranzistory*. Praha: SNTL - Nakladatelství technické literatury, 1979.
- [Win] Jingga Mutiara Windyarahma. *Save Data Input to SPIFFS with ESP32 and Arduino IDE*. URL: <https://medium.com/@ladolcefarniente/save-data-input-to-spiffs-with-esp32-and-arduino-ide-9e37011d7a91>.

Seznam obrázků

2.1	Schéma činností koncového zařízení	8
2.2	Schéma zapojení centrály	9
2.3	Schéma zapojení relé a diody	9
2.4	Schéma zapojení koncového zařízení s fotorezistorem v nepájivém kontaktním poli . . .	11
2.5	Schéma zapojení koncového zařízení se senzorem světla v nepájivém kontaktním poli .	11
3.1	Přístup ke kartě vlastnosti	15
3.2	Nastavení na kartě vlastnosti	16
3.3	Přístup k Manažéru desek	16
3.4	Instalace desky v prostředí manažéru	17
3.5	Zvolení zařízení ze seznamu	17

Seznam tabulek

2.1	Kalkulace nákladů	12
-----	-----------------------------	----