

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: Bc. Emil Miler

Praha, 2021



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Robot Karel
Třída: 4.Z
Školní rok: 2020/2021
Platnost zadání: 30. 9. 2021
Vedoucí práce: Šimon Schierreich

Název práce: **Název maturitního projektu.**

Pokyny pro vypracování:

Zde popište, čeho se chystáte v práci dosáhnout.

Doporučená literatura:

Zde bude vedoucím práce doplněna doporučená literatura.

URL repozitáře:

<https://gitlab.com/gjk/maturitni-projekt>

vedoucí práce

student

V Praze dne 10. 3. 2021

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 16. března 2021

Vladislav Aulich

Poděkování

Poděkování.

Abstrakt

Abstrakt.

Klíčová slova

klíčové slovo, další pojem, jiný důležitý termín, a ještě jeden

Abstract

Abstract.

Keywords

keyword, important term, another topic, and another one

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	8
2.3	Výroba prototypu	10
2.3.1	Kalkulace nákladů	10
3	Technická dokumentace	11
3.1	Ukázka sekce	11
3.1.1	A jedné podsekce	11
3.2	A další sekce	11
	Závěr	13
	Seznam použité literatury	15
	Seznam obrázků	17
	Seznam tabulek	18

1. Teoretická část

V první části maturitní práce by se měla objevit informace o tom, jaký problém řešíte. Co si Váš projekt klade za cíl?

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, 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 na tzv. „na dálku“. Součástí řešení musí být uživatelské rozhraní, možnost další automatizace a možnost ovládání „offline“.

2. Implementace

Druhá kapitola obsahuje detailní informace o tom, jak probíhala implementace. Zde se objeví zdůvodnění výběru technologií, řešení problémů, na které jste narazili, informace o použitých knihovnách apod. Pochvalte se, nikdo to za Vás neudělá. Přiznejte chyby, není to ostuda.

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 a to hned z několika důvodů. Čip má integrovanou wifi, k dispozici je velké množství dokumentace, hardwaru s příklady a knihovnamí. 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 ESPAsyncWebServer¹. 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 WiFi².

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.

¹<https://github.com/me-no-dev/ESPAsyncWebServer>

²<https://github.com/arduino-libraries/WiFi>

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

Pro zjednodušení přístupu k uživatelskému rozhraní je na zařízení spuštěn mDNS server na adrese <http://esp32.local>. Tato funkce je realizována pomocí knihovny ESPmDNS³.

Uživatelské rozhraní je vytvořeno pomocí HTML. HTML soubor je uložen ve vnitřní flash paměti čipu ESP. K ovládání filesystému SPIF je užito knihovny SPIFFS⁴. 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 modulů. 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 odchytlí 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 RCswitch⁵, 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).

³<https://github.com/espressif/arduino-esp32/tree/master/libraries/ESPmDNS>

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

⁵<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 CSVparser⁶. 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

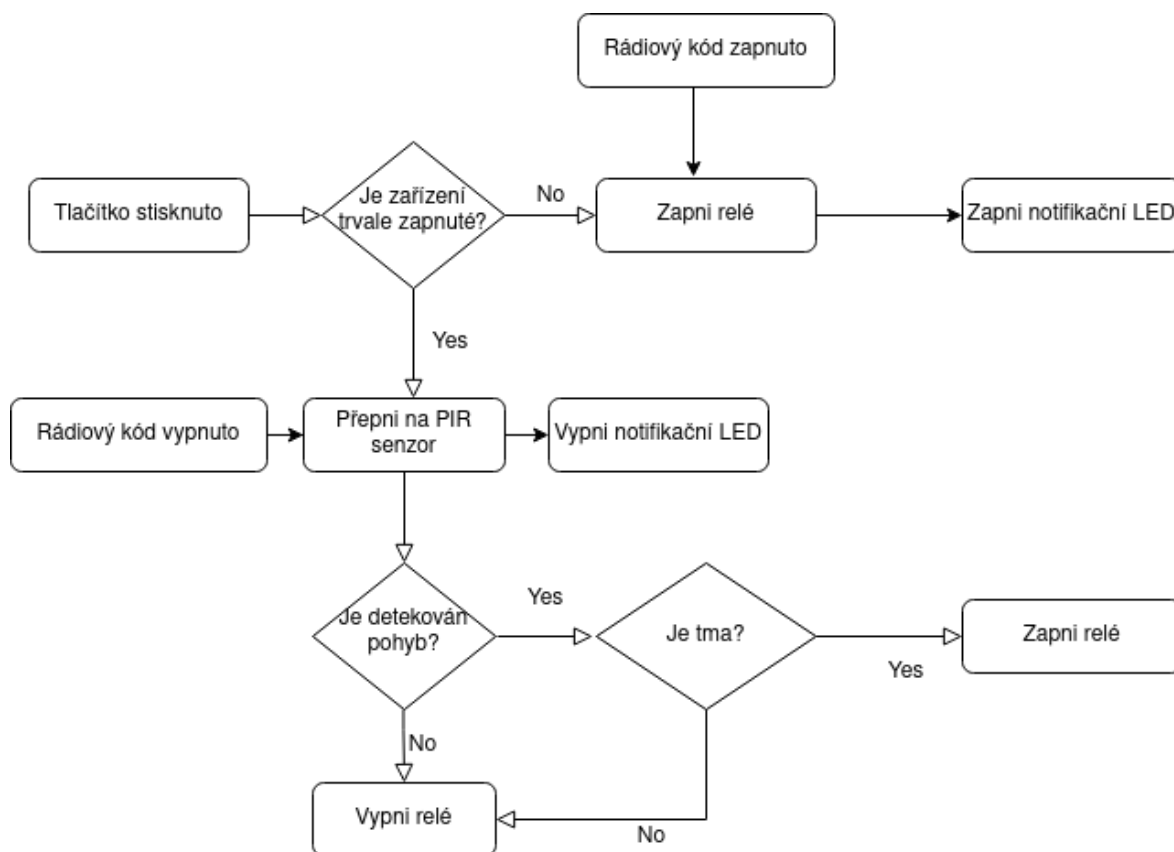
Pro „koncová zařízení“ jsem zvolil jako vývojovou platformu Arduino uno. Po odladění kódu a hardwaru jsem vyrobil prototyp, který již využíval Arduino Pro mini.

Pro realizaci zapínání a vypínání spotřebičů jsem zvolil ovládání pomocí relé. Pro snadnější ovládání jsem vybral již hotový modul pro arduino⁷, který pokud je na vstupu logická 1 vypnutý, pokud je přítomna zem, tak je zapnutý (je tzv. negovaný).

Reakce na pohyb je zajišťována PIR senzorem. 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.

⁶<https://github.com/michalmonday/CSV-Parser-for-Arduino>

⁷<https://dratek.cz/arduino/2954-modul-rele-5v-1-kanal-opticky-oddeleno.html>



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.

Obrázek Schéma zapojení

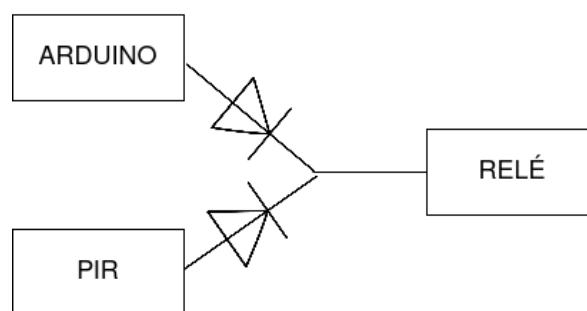
2.2.2 Koncové zařízení

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.

Toto řešení nefungovalo perfektně, kvůli úbytku napětí na diodě (0.7 V), takže výstup s PIR nebyl

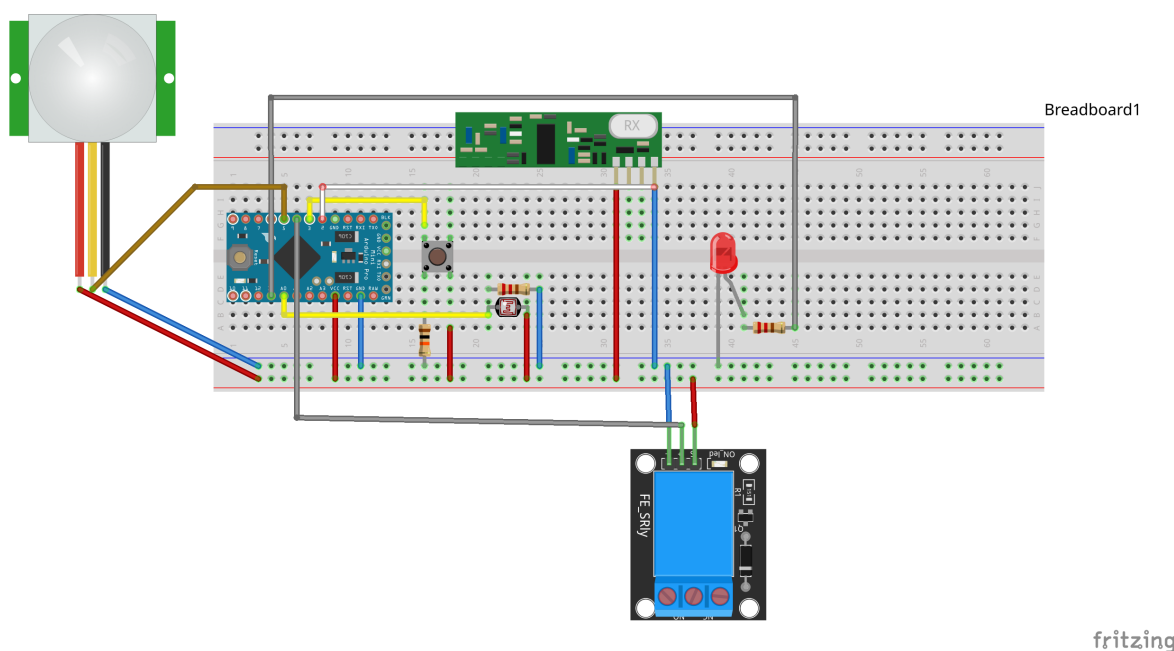


Obrázek 2.2: Schéma zapojení relé a diody

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.



Obrázek 2.3: Schéma zapojení v nepájivém kontaktním poli

Opět jsem vytvořil návrh krabičky pro 3D tisk. Při návrhu krabičky pro PIR senzor jsem vycházel již z hotového modelu⁸ staženého z portálu thinkiverse, 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.

⁸<https://www.thingiverse.com/thing:2845890>

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, ale myslím, že jsem nakonec udělal co bylo v mých silách.

2.3.1 Kalkulace nákladů

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 detektro pohybu	38

Tabulka 2.1: 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.

3. Technická dokumentace

Poslední kapitola obsahuje informace o tom, jak projekt, který v rámci maturitní práce vznikl, nainstalovat, spustit a používat.

3.1 Ukázka sekce

3.1.1 A jedné podsekce

3.2 A další sekce

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.

Seznam použité literatury

- [Ein05] Albert Einstein. "Zur Elektrodynamik bewegter Körper". In: *Annalen der Physik* 322.10 (1905), s. 891–921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.
- [Mal17] Martin Malý. *Hradla, volty, jednočipy: úvod do bastlení*. Praha: CZ.NIC, 2017. ISBN: 978-80-88168-23-2.
- [Mil87] Boleslav Černoch Milan Syrovátko. *Hradla, volty, jednočipy: úvod do bastlení*. Praha: SNTL - Nakladatelství technické literatury, 1987.
- [Old00] 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.

Seznam obrázků

2.1	Schéma činností koncového zařízení	8
2.2	Schéma zapojení relé a diody	9
2.3	Schéma zapojení v nepájivém kontaktním poli	9

Seznam tabulek

2.1	Kalkulace nákladů	10
-----	-----------------------------	----