

Střední průmyslová škola polytechnická,
Centrum odborné přípravy Zlín

Systém LED osvětlení „Microleaf Lygths“

Ročníková práce z oboru Informační technologie, se zaměřením na počítačové sítě a robotiku

Prohlašuji, že jsem tuto práci napsal samostatně.

Ve Zlíně dne: 20. června 2023

Obsah

1	Úvod	1
2	Základní pojmy	2
2.1	Hardware	2
2.2	Software	2
2.3	Mikrokontroléry	2
2.3.1	Von Neumannova architektura	3
2.3.2	Harvardská architektura	3
2.4	JavaScript	3
2.5	Node.js	4
2.6	LED	4
2.6.1	Barvy	4
2.7	Adresovatelný LED pásek	4
2.8	3D tisk	4
2.9	Application Programming Interface	5
3	Analýza Systému	7
3.1	Technická Proveditelnost	7
3.2	Požadavky zdrojů	8
3.3	Rizika a Omezení	9
4	Design	10
4.1	Mobilní Aplikace	10
4.1.1	Charakteristika dobře vytvořené mobilní aplikace	10
4.1.2	Klíčové funkce a požadavky aplikace	10
4.1.3	Vzhledu uživatelského rozhraní	10
4.2	Mikrokontrolér a komunikace	12
4.2.1	Instalace zařízení	12
4.2.2	Zpracování požadavků API	12
4.2.3	Adaptér sloužící pro konvertování dat	13
4.3	Realizace hardwaru	14
5	Závěr	15
6	Použité zdroje	16
7	Seznam Obrázků	18

1 Úvod

Tato ročníková práce se zaměřuje na vývoj mobilní aplikace, která uživatelům slouží jako digitální ovladač pro řízení světel na dálku. Cílem práce je poskytnout praktické řešení pro ty, kteří chtějí digitalizovat svou domácnost, ať už to jsou světla, nebo jiná zařízení.

Aplikace byla navržena tak, aby fungovala s konkrétním systémem, a obsahovala řadu přizpůsobitelných nastavení, která uživatelům umožňují vytvořit si prostředí osvětlení odpovídající jejím specifickým potřebám a preferencím.

Práce se zaměřila na různé aspekty vývoje mobilních aplikací, včetně návrhu uživatelského rozhraní, komunikací mezi aplikací a kontrolerem, a vývojem kontroleru samotného. Na začátku byly zvažovány klíčové vlastnosti a funkce, které by měl systém obsahovat, a jak je nejlépe integrovat. Poté bylo vše potřebné naprogramováno a hardwarově zrealizováno. Nezbytnou součástí byly provedeny všechny potřebné testy.

2 Základní pojmy

2.1 Hardware

Obecné označení pro zařízení, které tvoří počítač. Skládá se ze samotného počítače, vstupních zařízení, jako je klávesnice, myš a skener, úložných zařízení, jako je pevný disk, a výstupních zařízení, jako je displej. Naproti tomu programy, které zajišťují práci počítače, se nazývají software. V počítači je hardware řízen softwarem a oba tyto prvky spolu dobře spolupracují a fungují jako systém [1].

2.2 Software

Softwarem rozumíme veškeré programové vybavení, které může být na počítači provozováno.

Běžně se můžeme setkat s rozdělením software na systémový software (Operační systém) a aplikační software (ostatní programy). Systémový software je v podstatě speciální médium, které zprostředkovává komunikaci mezi hardwarem a konkrétní aplikací. Mezi systémový software tedy patří firmware, BIOS a Operační systém. Bez dříve jmenovaných by nebylo možné druhou velkou skupinu, aplikační software, provozovat. Aplikačním software máme tedy na mysli rozličné programy sloužící k nějakému konkrétnímu účelu [2]. Může se jednat o programy určené k ochraně a údržbě počítače a dat, například antivirové nebo zálohovací programy. Dále textové editory, různorodé aplikace sloužící ke vzájemné komunikaci s okolním světem, grafické editory, hry, ale i internetové prohlížeče.

2.3 Mikrokontroléry

Jednočipový počítač nebo také angl. Microcontroller (Mikrokontrolér, MCU, μC) je většinou monolitický integrovaný obvod obsahující kompletní mikropočítač.

Jednočipové počítače se vyznačují velkou spolehlivostí a kompaktností, proto jsou určeny především pro jednoúčelové aplikace jako je řízení, regulace apod. Často jsou jednočipové počítače součástí vestavěných (embedded) systémů.

Jednočipový počítač je integrovaný obvod, který v sobě zahrnuje zpravidla vše potřebné k tomu, aby mohl obsáhnout celou aplikaci, aniž by potřeboval další podpůrné obvody. Především jde o paměť pro uložení programu (FLASH, EEPROM nebo ROM) a operační paměť RAM pro uložení aplikačních proměnných a zásobníku. Většina jednočipových počítačů také obsahuje rozsáhlou sadu podpůrných obvodů. Typicky jde o bloky pro logické a analogové vstupy/výstupy, pro komunikační linky, pro rozdělení strojového času a další aplikační logiku.

Existují základní dvě architektury mikroprocesorů, a to von Neumannova a Harvardská, každá má svoje výhody i nevýhody. Při současném stupni integrace se zřejmě častěji využívá Harvardská architektura, vysoký stupeň integrace dovoluje připojit různé bloky paměti pomocí vlastních sběrnic.

Dělení na „Harvardskou“ a „Von Neumannovu“ architekturu je však při dnešním stupni integrace již poněkud akademické. U moderních architektur se často uživateli adresový prostor

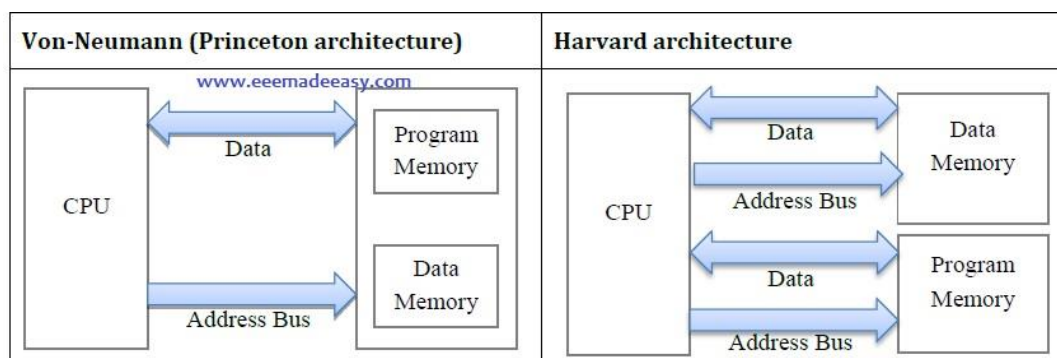
jeví navenek jako lineární (Von Neumannovský), zatímco fyzicky jsou paměti k jádru připojeny pomocí několika nezávislých sběrnic (např. jedna sběrnice pro FLASH/ROM, druhá pro uživatelskou vnitřní RAM a zásobník, třetí pro připojení integrovaných paměťově mapovaných periférií, další pro připojení externí RAM) [3].

2.3.1 Von Neumannova architektura

Je architektura, pro kterou je typická společná paměť pro data i program. Toto uspořádání má výhody v tom, že nepotřebujeme rozlišovat instrukce pro přístup k paměti dat a paměti programu, což vede k zjednodušení vlastního čipu. Další výhodou je, že je potřeba pouze jedna datová sběrnice, po které se přenáší oba typy dat, což je výhodné v případě použití externích pamětí, kdy se redukuje potřebný počet nutných vstupů a výstupů. Nevýhodou je, že přenos obou typů dat po jedné sběrnici je pomalejší, než při oddělených sběrnicích pro paměti dat a programu [3].

2.3.2 Harvardská architektura

Je typická oddělením paměti programu a paměti dat. Hlavní nevýhodou této architektury je větší technologická náročnost daná nutností vytvořit dvě sběrnice. Za hlavní výhodu lze považovat možnost jiné šířky programové a datové sběrnice. Této možnosti se široce využívá, takže najdeme osmibitové mikrokontroléry s programovou sběrnicí širokou 12, 14 i 16 bitů. Mezi další výhody harvardské architektury patří rychlost vykonávání instrukcí, protože instrukci i potřebná data lze číst v jeden okamžik [3].



Obr. 1 Diagramy architektur mikrokontrolérů

2.4 JavaScript

JavaScript je dynamický programovací jazyk. Je lehký a nejčastěji se používá jako součást webových stránek, jejichž implementace umožňuje skriptům na straně klienta komunikovat s uživatelem a vytvářet dynamické stránky. Jedná se o interpretovaný programovací jazyk s objektově orientovanými možnostmi.

JavaScript byl nejprve známý jako LiveScript, ale společnost Netscape změnila jeho název na JavaScript, pravděpodobně kvůli vzrušení, které vyvolala Java. JavaScript se poprvé objevil v Netscape 2.0 v roce 1995 pod názvem LiveScript. Obecné jádro jazyka bylo zabudováno do prohlížečů Netscape, Internet Explorer a dalších webových prohlížečů [4].

2.5 Node.js

Node.js je prostředí umožňující spouštět JavaScript kód mimo webový prohlížeč. Je postaveno na JavaScript enginu Chrome V8, takže základ tohoto JS prostředí je stejný jako ve webovém prohlížeči Google Chrome [5]. Důležité je říct, že i přes to, že v tomto prostředí lze psát cokoliv, jeho návrh a primární účel je tvorba serverové části webových aplikací. To má společné například s jazykem PHP, který má stejné zaměření.

JavaScript se díky tomuto prostředí dá používat i na serveru a ne pouze na druhém konci, u klienta. Avšak na rozdíl od zmíněného PHP je v Node.js kladen důraz na vysokou škálovatelnost, tzn. schopnost obsloužit mnoho připojených klientů současně. Pro tuto vlastnost a vysokou výkonnost je dnes Node.js velmi oblíbený pro tvorbu tzv. API serverů pro klientské single page aplikace rovněž v JavaScriptu [5].

2.6 LED

Světelná dioda, LED, je polovodičové zařízení, které po průchodu elektrického proudu vyzařuje světlo. Světlo vzniká, když se částice, které přenášejí proud, spojí v polovodičovém materiálu.

Vzhledem k tomu, že světlo vzniká v pevném polovodičovém materiálu, jsou LED diody označovány jako polovodičová zařízení. Termín polovodičové osvětlení, který zahrnuje také organické diody LED, OLED. Odlišuje tuto technologii osvětlení od jiných zdrojů, které používají žhavená vlákna, žárovky a wolframové halogenové žárovky, nebo plynové výboje, zářivky [6].

2.6.1 Barvy

Uvnitř polovodičového materiálu LED jsou elektrony a díry obsaženy v energetických páslech. Rozdělení pásů určuje energii fotonů, které LED vyzařuje.

Energie fotonů určuje vlnovou délku vyzařovaného světla, a tím i jeho barvu. Různé polovodičové materiály s různými pásovými mezerami vytvářejí různé barvy světla. Přesnou vlnovou délku, barvu, lze vyladit změnou složení oblasti vyzařující světlo neboli aktivní oblasti [6].

2.7 Adresovatelný LED pásek

Individuálně adresovatelný LED pásek se také nazývá digitální LED pásek, pixelový LED pásek, magický LED pásek nebo vysněný barevný LED pásek, je LED pásek s řídicími IC, které umožňují ovládat jednotlivé LED diody nebo skupiny LED diod. Můžete ovládat konkrétní část led pásku, proto se mu říká „adresovatelný“ [7].

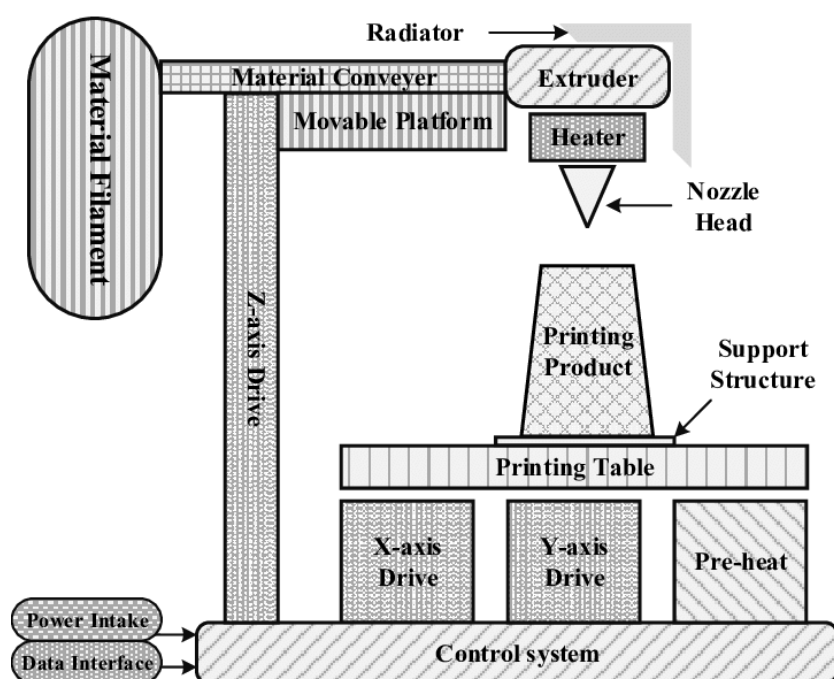
2.8 3D tisk

3D tisk, nebo-li aditivní výroba, je proces výroby trojrozměrných pevných objektů z digitálního souboru.

Vytvoření 3D tištěného objektu se dosahuje pomocí aditivních procesů. Při aditivním procesu se objekt vytváří kladením postupných vrstev materiálu, dokud není objekt vytvořen. Každou z těchto vrstev si lze představit jako tenký řez objektem.

Je opakem subtraktivní výroby, která spočívá ve vyřezávání / vydlabávání kusu kovu nebo plastu například pomocí frézky. Umožňuje vyrábět složité tvary s použitím menšího množství materiálu než tradiční výrobní metody [8].

Dnes se pomocí tohoto tisku dá vytisknout v podstatě cokoliv, co má tvar. Od nenáročných modelů, jako třeba prstýnek, až po obyvatelný dům či jiné struktury.



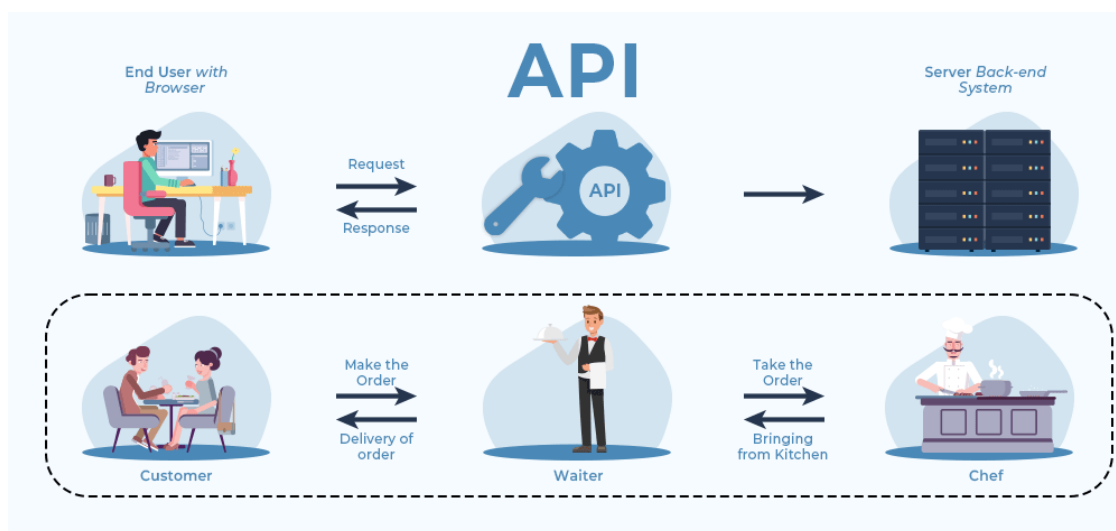
single
speci
extra
newe
addre
revie
addit
meas
impo
mate
Al
Emis
inter
impa
with
fram
of e
syste
annr

Obr. 2 Schématický diagram 3D tiskárny

2.9 Application Programming Interface

API neboli rozhraní pro programování aplikací je soubor definovaných pravidel, která umožňují různým aplikacím vzájemnou komunikaci. Funguje jako zprostředkující vrstva, která zpracovává přenosy dat mezi systémy a umožňuje společností otevřít data a funkce svých aplikací externím vývojářům třetích stran, obchodním partnerům a interním oddělením ve společnosti.

Definice a protokoly v rámci rozhraní API pomáhají firmám propojit mnoho různých aplikací, které používají při každodenním provozu, což šetří čas zaměstnanců a odbourává síla, která brání spolupráci a inovacím. Vývojářům poskytuje dokumentace API rozhraní pro komunikaci mezi aplikacemi, což zjednodušuje jejich integraci [9].



Obr. 3 Ilustrace funkce API

3 Analýza Systému

3.1 Technická Proveditelnost

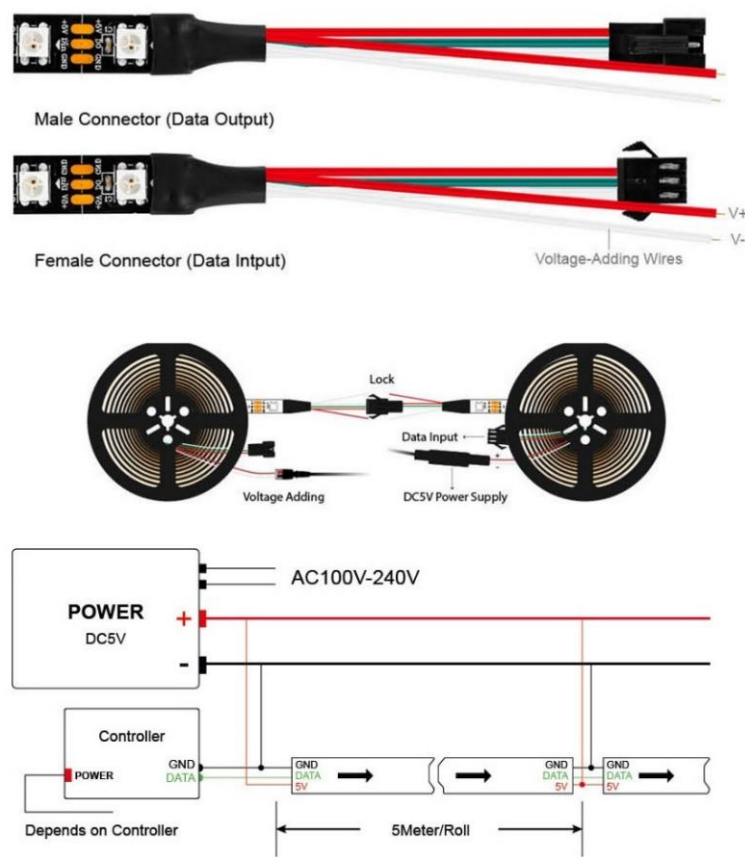
Použití frameworku React Native pro vývoj mobilní aplikace nabízí několik výhod, včetně kompatibility napříč platformami a možnosti využívat předem připravené komponenty a knihovny. Tato volba frameworku umožnila snadno vytvořit uživatelsky přívětivé rozhraní pro ovládání světel.

Komunikace mezi aplikací a kontrolerem byla provedena prostřednictvím volání požadavků API, zasílaných na webový server, který je automaticky spuštěn v místní síti Wi-Fi, ve které se mikrokontroler nachází. Tento přístup umožňuje jednoduchý a efektivní způsob ovládání systému prostřednictvím aplikace.

Protože se tato práce zaměřuje i na co největší volnost uživatele jak světla bude používat, práce se inspirovala již oficiálně produkovánými světly značky Nanoleaf. Ty byly v poslední době, kdy tato práce byla napsána, velice populární, a to hlavně mezi lidmi, kteří se živí vysíláním živého přenosu na různých webových platformách, jako je Twitch nebo YouTube. Nanoleaf světla jsou díky svému hexagonovému či trianglovému tvaru vcelku modulární, protože si je uživatel může umístit do tvaru, jakého chce.

Jako zdroj světla byly použity LED pásy, které byly rozděleny do navržených 3D modelů. Ty se poté propojily a zajistilo se, aby každá LED dioda dostala napětí, které potřebuje.

Hardwarové komponenty použité v projektu, včetně LED pásků, tištěných modelů a mikrokontroleru, se zdály jako vhodné pro požadavky práce. Vybraný mikrokontroler nabízí potřebný výpočetní výkon a možnosti připojení Wi-Fi i Bluetooth, pro podporu komunikace mezi mobilní aplikací a kontrolerem. LED pásy poskytují spolehlivý a energeticky úsporný způsob osvětlení uvnitř trojúhelníků.



Obr. 4 Schéma propojení LED pásků s mikrokontrolérem

Aby byla zajištěna technická proveditelnost projektu, bylo potřeba důkladně otestovat a zajistit, aby aplikace a systém fungoval tak, jak má. To zahrnovalo i ověření stability a škálovatelnosti.

Závěrem lze říct, že technická proveditelnost práce se jevila jako vysoká. Použití zavedených frameworků a technologií, jako je React Native, API, a dostupnost vhodných hardwarových komponent, jako je mikrokontroler a LED pásky, naznačují, že projekt jde úspěšně vyvinout a realizovat.

3.2 Požadavky zdrojů

Vývoj mobilní aplikace a světél vyžaduje řadu zdrojů, včetně hardwaru, softwaru a autora, který na práci bude pracovat.

Hardwarové požadavky projektu zahrnují LED pásky, které dokážou přijímat datové signály. Pro tuto práci byly vybrány pásky Neopixel WS2812B, které jsou volně dostupné k nákupu a splňují vše, co vyžadujeme. Dále je potřeba mikrokontrolér, který dokáže sloužit jako hlavní řídicí jednotka pro komunikaci a zasílání potřebných dat a napětí do světél. Jako vhodný kandidát se zdál ESP8266, který nabízí i integrovaný Wi-Fi modul. Pro tisknutí 3D modelů je také zapotřebí tiskárna, která byla se shodou okolností dostupná v učebně

informatiky, na Střední průmyslové škole polytechnické ve Zlíně. Tiskárna se nazývá Prusa i3, která splnila všechny velikostní požadavky 3D modelů.

Softwarové požadavky projektu zahrnují framework React Native, který k vývoji mobilní aplikace využívá JavaScript framework, Node.js. Ke komunikaci mezi mobilní aplikací a kontrolerem slouží lokální webový server, který v praxi přijímá a popřípadě odesílá potřebná data. Pro vývoj je nezbytná znalost pokročilejšího programování. Pro React Native znalost JavaScriptu a principu Node.js, a C++, ve kterém se často odehrává vývoj hardwaru. Potřeba bude také software pro správu verzí, například Git, a editor kódu, Visual Studio Code, Atom, či jiné volně dostupné editory.

Nároky na zdroje jsou pro tuto práci mírné, přičemž hardwarové komponenty a softwarové nástroje jsou snadno dostupné a cenově přijatelné.

3.3 Rizika a Omezení

Stejně jako u každého projektu vývoje softwaru existují potenciální rizika a omezení, která mohou ovlivnit průběh či úspěch vývoje celkového systému.

Jedno z potenciálních rizik se týká kompatibility aplikace s různými zařízeními a operačními systémy. Použití React Native sice může pomoci toto riziko zmírnit, ale pro zajištění kompatibility je důležité aplikaci důkladně otestovat na řadě zařízení a platform.

Další potenciální riziko souvisí se zabezpečením systému. Vzhledem k tomu, že aplikace a kontroler spoléhají na komunikaci prostřednictvím místní sítě Wi-Fi, existuje riziko neoprávněného přístupu nebo rušení. Na alespoň z části kompletní eliminaci tohoto rizika ale není dostatek času, ani zkušených lidí, proto je spoléháno na základní bezpečnost knihoven, které tuto komunikaci spravují.

Omezení práce souvisí i s hardwarovými součástmi, použitými v tomto systému. LED pásy, či jednotlivé diody, a tištěné trojúhelníky, sice nabízejí flexibilní a přizpůsobitelné možnosti, ale mohou být omezeny svou fyzickou velikostí a konstrukcí. To může mít dopad na škálovatelnost a rozsah možností pro koncové uživatele.

Pro zmírnění těchto rizik a omezení by měl vývoj upřednostnit důkladné testování a pečlivou analýzu potřeby hardwaru, aby nedošlo ke zkratu nebo horším následkům. Je důležité být flexibilní a přizpůsobivý a podle potřeby provádět úpravy, aby byla práce úspěšně dokončena.

4 Design

4.1 Mobilní Aplikace

4.1.1 *Charakteristika dobře vytvořené mobilní aplikace*

Mobilní aplikace mají vysokou míru odchodu, takže pro udržení zájmu uživatelů je zásadní, aby na ně aplikace udělala skvělý první dojem. Tento první dojem obvykle začíná intuitivním a poutavým uživatelským rozhraním, tedy klíčovou vlastností, kterou je třeba mít na paměti při vytváření aplikace.

Dobře navržené uživatelské rozhraní se týká jak vzhledu a dojmu z aplikace, tak skutečných funkcí. I když aplikace poskytuje uživateli velkou hodnotu, pokud není intuitivní, uživatele rychle ztratíte, protože nebudou investovat čas do učení se rozhraní. A pokud nebude vizuálně přitažlivá, uživatelé se jí nebudou věnovat, což ztíží její přijetí v celé organizaci.

Většina lidí používá aplikace na mobilním zařízení, takže uživatelské rozhraní by mělo být optimalizováno pro malou dotykovou obrazovku. To znamená odstranit všechny nepodstatné funkce aplikace, aby design nepůsobil příliš rušně nebo nepřehledně, a také zajistit konzistenci designu na různých platformách a velikostech telefonů.

Konzistence designu je nutná také u typografie, tlačítek, ikon a dalších prvků značky. Uživatelé tak získají jednodušší dojem při procházení aplikací a navíc se zlepší jejich čitelnost. Struktura aplikace by měla být také konzistentní, přičemž nejdůležitější obsah by měl mít větší vizuální váhu, pomáhá totiž s hierarchií informací, usnadňuje uživatelský tok a zlepšuje uživatelský zážitek [10].

4.1.2 *Klíčové funkce a požadavky aplikace*

Před zaměřením na design aplikace a kódu samotného, je potřeba určit, co přesně od této aplikace vyžadujeme. Ta by měla sloužit jako digitální ovladač, místo ovladače fyzického. Základní funkce fyzického ovladače, který dnes skoro vždy dostanete k LED páskům zakoupených u výrobce, je změna vlnové délky elektromagnetického záření v diodách, bez té by si pásek skoro nikdo nezakoupil. Dále v něm můžeme najít změnu citlivosti a různé animace, ty jsou ale ve většině případů vážně špatné a nepříjemné na pohled.

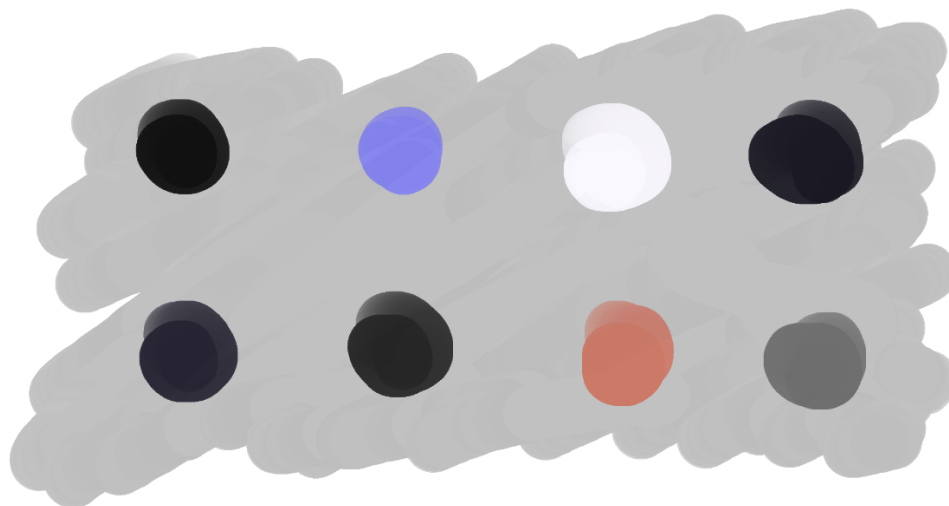
Pokud jsou zahrnuty všechny tyto funkce, tedy změna barev, citlivost světla a pokus o příjemnější animace, je zde vcelku pevný základ, na kterém lze začít. Ale vzhledem k tomu, že světla jsou v plánu zrealizovat modulárně, nelze samotnou modularitu vynechat. Pro dosažení tohoto cíle byla zavedena do základních funkcí jedna položka, a to možnost změny jednotlivých modulů. To znamená, že uživatel má volnost různých kombinací barev a citlivostí, podle jeho preferencí. Poslední přidaná hodnota je možnost přepnutí mezi různými mikrokontroléry, díky které koncový uživatel má možnost ovládat více vlastněných systémů stejné struktury pomocí jedné mobilní aplikace.

4.1.3 *Vzhledu uživatelského rozhraní*

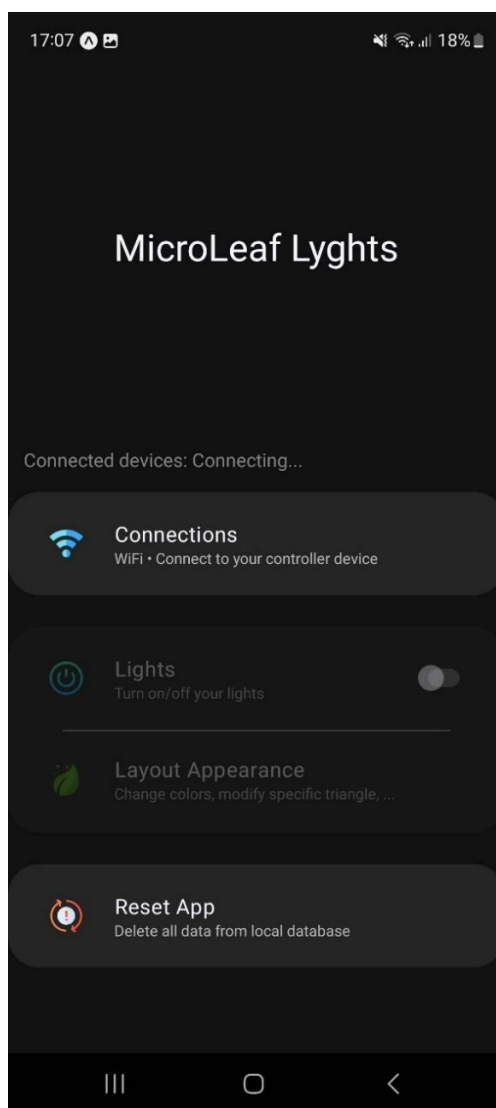
Aplikace je navržena tak, aby se v ní dalo lehce vyznat, proto bylo zavedeno pravidlo, které určuje, že uživatel nejde o dál jak 2 kroky. To v praxi znamená, aplikace obsahuje jednu hlavní

stranu, která uživatele navádí do ostatních částí aplikace, jako je výběr konkrétního aktivního systému nebo ovladač systému. Pokud je třeba informovat uživatele, nebo zobrazit obsah, který není stálý, je využito vyskakovací okno, které lze lehce pomocí jednoho kliku minimalizovat.

Aby bylo rozhraní i oživeno a uživateli příjemně, byly použity barvy tmavšího typu a ikony, které byly čerpány z nativní knihovny React a veřejně dostupné webové stránky flaticon.



Obr. 5 Seznam použitých barev v aplikaci



Obr. 6 Hlavní strana aplikace

4.2 Mikrokontrolér a komunikace

4.2.1 Instalace zařízení

Při instalaci zařízení do domácnosti či jiného zařízení je třeba mikrokontrolér uvést základní údaje o místní síti a počtu led, který se vypočítá pomocí vynásobením počtu nainstalovaných modulů šesti.

Po zadání základních údajů se data nahrají do mikrokontroléru, který se následně automaticky připojí do místní sítě během 30 sekund, ihned po dodání elektřiny a potřebného napětí.

4.2.2 Zpracování požadavků API

Aby mikrokontrolér mohl přijímat tyto požadavky, musí být připojen k síti. K té se připojí pomocí knihovny *ESP8266WiFi*, která umožňuje připojení, a knihovny *SPI*, ta spravuje Wi-Fi čip, jenž je součástí *ESP8266* mikrokontroléru. Jakmile dojde k připojení do sítě s alespoň

minimální ochranou, lze IP adresu mikrokontroléru a informace o připojení či jiných zprávách získat pomocí *Serial Monitor*, nebo v aplikaci či webu správy routeru místní sítě.

Požadavky přijaté z aplikace jsou zpracovány následujícím způsobem. Kód v mikrokontroléru data z požadavků rozdělí na řádky, které postupně registruje a rozvětjuje do částí softwaru. Po tomto procesu využije vytvořených knihoven, za účelem rozeslání dat do ostatních knihoven, které mají za úkol data konvertovat a zaslat do jednotlivých LED.

4.2.3 Adaptér sloužící pro konvertování dat

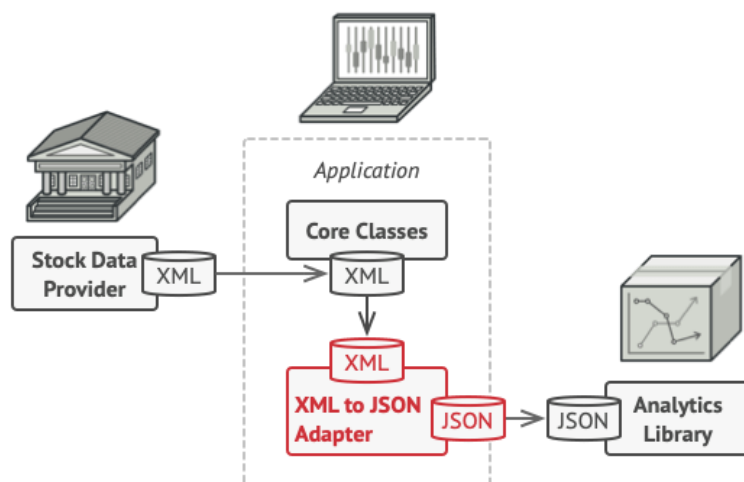
Data přijatá z aplikace jsou potřeba překonvertovat do aktivní architektury hardwaru, kterou koncový uživatel vlastní. Pomocí adaptéru, který byl vytvořen pro tento účel, jsou data jako modul, barvy, nebo animace převáděny na jednotlivé signály pro sestavený hardware. Pokud tedy například mikrokontrolér získá požadavek o změnu barvy, adaptér si jako parametr vezme danou barvu, překonvertuje ji do potřebného formátu a pošle ji do jednotlivých LED, kterým je barva určena. Pokud v této situaci uživatel neoznačil žádný modul, ve kterém by se barva měla změnit, program akci přeskočí. Naopak animace nebo vypnutí zařízení jsou spravovány globálně, proto není třeba zadávat jednotlivý sektor jako součást požadavku.

Konvertovaná data jsou zaslána do knihovny FastLED, která rozesílá signály do jednotlivých diod.

4.2.3.1 Co je to Adaptér?

Vzor adaptéru funguje jako most mezi dvěma nekompatibilními rozhraními. Tento typ návrhového vzoru spadá pod strukturální vzor, protože tento vzor kombinuje schopnosti dvou nezávislých rozhraní.

Tento vzor zahrnuje jedinou třídu, která je zodpovědná za spojení funkcí nezávislých nebo nekompatibilních rozhraní. Reálným příkladem může být čtečka karet, která funguje jako adaptér mezi paměťovou kartou a notebookem. Paměťovou kartu připojíte ke čtečce karet a čtečku karet k notebooku, aby bylo možné paměťovou kartu číst prostřednictvím notebooku [11].



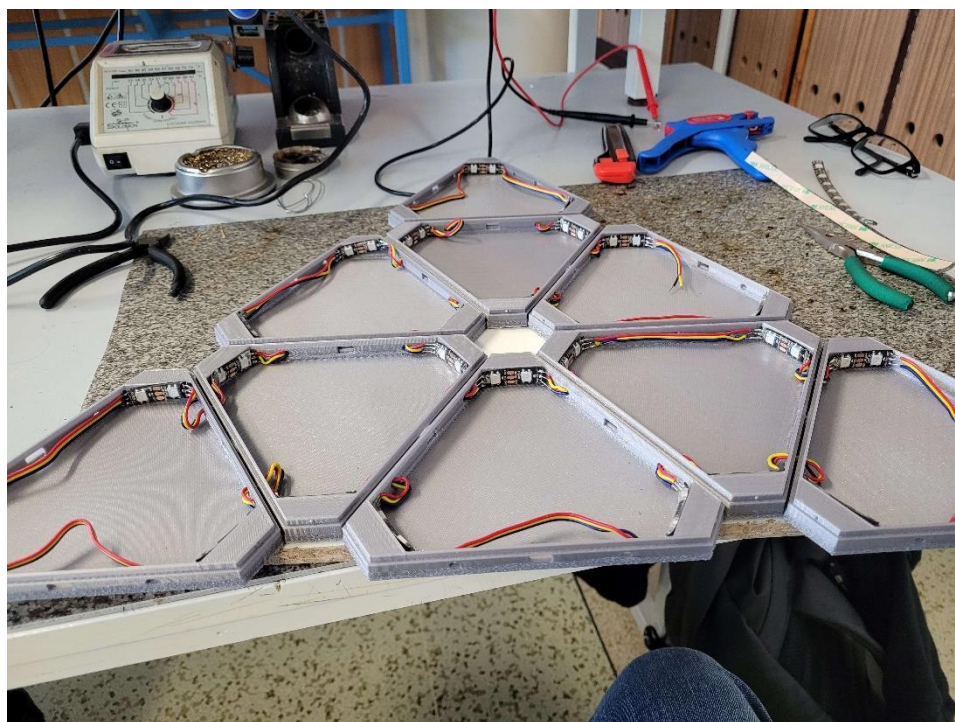
Obr. 7 Účel Adaptéru v praxi

4.3 Realizace hardwaru

Tato práce byla inspirována společností Nanoleaf, která podobný typ světél již oficiálně vyrábí. Aby tedy byla kompletní a nezaměřovala se pouze na software, byl přidán i plně funkční hardware, který mikrokontrolér spravuje.

Byly využity veřejně dostupné modely. Tyto modely byly stáhnuty a následně vytisknuty na tiskárně Prusa i3. Jsou k nalezení ve zdrojích tohoto dokumentu. Bylo vytištěno 9 kosených trojúhelníků, přičemž každý obsahuje tři pásy po dvou diodách, které jsou později navázány k sobě na straně softwaru.

Aby bylo možno diody ovládat, musí být propojeny vodičem, který dokáže vést data. Dohromady tedy tři vodiče sériově propojují diody, které vedou od mikrokontroléru celou konstrukcí, což vede k pevnému zapojení. To bylo zváženo, ale kvůli časové i materiálové náročnosti plné modularity, nelze moduly spojit jinak, než napevno.



Obr. 8 Hardwarový proces realizace světél

5 Závěr

Práce se zaměřovala na vývoj mobilní aplikace, která uživatelům slouží jako digitální ovladač, a realizaci hardwaru, tedy světel samotných. Za cíl si kladla poskytnout praktické řešení pro ty, kteří chtějí digitalizovat svou domácnost, ať už to jsou světla nebo jiná zařízení.

V úvodní kapitole byly definovány základní pojmy, které byly při práci potřeba znát. Poté byl cíl analyzován, což umožnilo jednodušší vývoj celého systému. Jakmile proběhla analýza práce, byl zahájen vývoj mobilní aplikace a komunikace mezi mikrokontrolérem a prototypem aplikace. V pozdější fázi vývoje bylo započato zrealizování konstrukce modulů a světel, které mikrokontrolér ovládal.

Při implementaci rozšíření změny barev a animací v aplikaci byla identifikována nesrovnalost barev, což znamenalo, že barva na displeji mobilního zařízení nebyla identická té promítané. Po detailnější identifikaci byl problém ignorován z důvodu zjištěných limitů diod, které dokáží vyzařovat omezený rozsah frekvencí. S aktuálním hardwarem je nereálné umožnit uživateli promítnout všech 16,777,216 barev, které jsou součástí spektra RGB, v odpovídajícím odstínu. Další limity byly objeveny u programování mikrokontroléru, jehož hardware neumožňuje spouštět kód ve více vláknech procesoru. Vzhledem k tomu, že na jednom vlákně je již spuštěn webový server, jediným způsobem jak přijmout kód podporovat animace spočívá v metodice, která limitovaně umožňuje spouštět více částí kódu na jednom vlákně. Díky tomu vzniká větší odezva při správě API požadavků, protože pokud je animace spuštěna, program zasílá data do jednotlivých diod.

I přes překážky a limity, které se v závěru vyskytly, systém byl úspěšně vyvinut. Umožňuje ovládat vzdálená zařízení pomocí digitálního ovladače, tedy aplikace na mobilním zařízení uživatele. Ve vypracované konfiguraci podporuje výběr ze seznamu zařízení připojené k místní síti, možnost digitálního označení sestavy modulů světel, změnu barev diod a základní animace.

6 Použité zdroje

- [1] コトバンク, “ハードウェア,” [オンライン]. Available: <https://kotobank.jp/word/ハードウェア-7166>. [アクセス日: 16 4 2023].
- [2] P. I. M. M. D. PaedDr. Karel Myška, „Základní hardware a software, operační,“ [Online]. Available: https://www.uhk.cz/file/edee/filozoficka-fakulta/studium/myska_-_zakladni_hardware_a_software_operacni_system.pdf.
- [3] isst.hys.cz, „MIKROKONTROLÉRY,“ [Online]. Available: <http://isst.hys.cz/images/prezentace/mikrokontrolery.pdf>. [Přístup získán 15 4 2023].
- [4] tutorialspoint, "JavaScript - Overview," tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/javascript/javascript_overview.htm. [Accessed 16 4 2023].
- [5] J. Máca, „Úvod do Node.js,“ ITNetwork, [Online]. Available: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs/>. [Přístup získán 15 4 2023].
- [6] ledsmagazine, „What is an LED?,“ [Online]. Available: <https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led>. [Přístup získán 16 4 2023].
- [7] ledyilighting, "The Ultimate Guide To Addressable LED Strip," LEDYi, [Online]. Available: <https://www.ledyilighting.com/the-ultimate-guide-to-addressable-led-strip/>. [Accessed 16 4 2023].
- [8] 3dprinting, "What is 3D Printing?," 3dprinting, [Online]. Available: <https://3dprinting.com/what-is-3d-printing/>. [Accessed 16 4 2023].
- [9] IBM, "What is an API?," IBM, [Online]. Available: <https://www.ibm.com/topics/api>. [Accessed 18 4 2023].
- [10] Microsoft, "Characteristics of a Good Mobile App," Microsoft Power Apps, 2023. [Online]. Available: <https://powerapps.microsoft.com/en-us/what-makes-a-good-app/>. [Accessed 15 4 2023].
- [11] Tutorialspoint, "Design Patterns - Adapter Pattern," Tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/design_pattern/adapter_pattern.htm. [Accessed 18 4 2023].
- [12] M. Abrams, "3D Printing Houses," ASME, [Online]. Available: <https://www.asme.org/topics-resources/content/3d-printing-houses>. [Accessed 16 4 2023].

- [13] AkshitaKumawat, „What is an API (Application Programming Interface)?“, GeeksForGeeks, [Online]. Available: <https://www.geeksforgeeks.org/what-is-an-api/>. [Přístup získán 16 4 2023].
- [14] D. Zhart, „Adapter,“ [Online]. Available: <https://refactoring.guru/design-patterns/adapter>. [Přístup získán 18 4 2023].
- [15] T. Peng, "A schematic diagram of a 3D printer," [Online]. Available: https://www.researchgate.net/figure/A-schematic-diagram-of-a-3D-printer_fig1_295241769. [Accessed 18 4 2023].
- [16] laskakit, „Led pásek Neopixel WS2812B 60led/m IP65 5m černý,“ [Online]. Available: <https://www.laskakit.cz/led-pasek-neopixel-ws2812b-60led-m-ip65-5m-cerny/>. [Přístup získán 18 4 2023].
- [17] Harvahammas, "Nanoleaf Light Panel Replica," [Online]. Available: <https://www.thingiverse.com/thing:3230905>. [Accessed 18 4 2023].

7 Seznam Obrázků

Obr. 1	Diagramy architektury mikrokontrolérů	3
Obr. 2	Schématický diagram 3D tiskárny	5
Obr. 3	Ilustrace funkce API.....	6
Obr. 4	Schéma propojení LED pásků s mikrokontrolérem	8
Obr. 5	Seznam použitých barev v aplikaci	11
Obr. 6	Hlavní strana aplikace	12
Obr. 7	Účel Adaptéru v praxi	13
Obr. 8	Hardwarový proces realizace světla	14