



## Zadání bakalářské práce

<b>Název:</b>	Řídicí jednotka pro individuálně adresovatelné led pásky s Raspberry Pi Zero
<b>Student:</b>	Miroslav Krýcha
<b>Vedoucí:</b>	Ing. Pavel Kubalík, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Počítačové inženýrství
<b>Katedra:</b>	Katedra číslicového návrhu
<b>Platnost zadání:</b>	do konce letního semestru 2022/2023

### Pokyny pro vypracování

- 1) Prozkoumejte existující řešení.
- 2) Navrhněte vlastní řešení řídicí jednotky pro připojení led pásky.
- 3) Řídicí jednotka bude splňovat tyto požadavky:
  - možnost připojení dvou a více led pásků,
  - podpora více typu led. Alespoň jeden 12V a 5V,
  - jednotku bude možno ovládat pomocí tlačítek, dálkového ovládání a SSH,
  - pro obsluhu k jednotce připojte Waveshare 2.7 e-ink display a vytvořte vhodné grafické rozhraní,
  - jednotka bude obsahovat senzor zvuku a světla sloužících k automatickému spouštění.
- 4) Navržené řešení zrealizujte a řádně otestujte.



Bakalářská práce

**ŘÍDICÍ JEDNOTKA PRO  
INDIVIDUÁLNĚ  
ADRESOVATELNÉ LED  
PÁSKY S RASPBERRY PI  
ZERO**

**Miroslav Krýcha**

Fakulta informačních technologií  
Katedra číslicového návrhu  
Vedoucí: Ing. Pavel Kubalík, Ph.D.  
10. května 2022

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2022 Miroslav Krýcha. Odkaz na tuto práci.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

Odkaz na tuto práci: Krýcha Miroslav. *Řídící jednotka pro individuálně adresovatelné led pásy s Raspberry Pi Zero*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

# Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratek	x
<b>1 Úvod</b>	<b>1</b>
<b>2 Popis problému, specifikace cíle</b>	<b>3</b>
2.1 Shrnutí funkčních požadavků	3
<b>3 Existující řešení</b>	<b>5</b>
3.1 Mini 21 Key RF/IR Remote	5
3.2 SP1008E	6
3.3 SP107E	6
3.4 K-1000C	7
<b>4 Teorie</b>	<b>9</b>
4.1 Ohmův zákon	9
4.2 Výkon elektrického pole	9
4.3 Úbytek napětí na vedení	9
4.4 Mooreův stroj	9
4.5 SSH	10
4.6 Pulzně šířková modulace	10
4.7 Průřez vodiče	10
<b>5 Analýza</b>	<b>11</b>
5.1 Výběr komponent	11
5.1.1 Raspberry Pi	11
5.1.2 LED pásy	12
5.1.3 Napájecí zdroje	14
5.2 Programovací jazyk a prostředí	15
5.2.1 Debug	15
<b>6 Návrh řešení</b>	<b>17</b>
6.1 Počáteční návrh	17
6.2 Pokročilý/finální návrh	18
6.2.1 Software	18
6.2.2 Hardware	19
6.3 Uživatelská rozhraní	19

<b>7</b>	<b>Realizace softwaru</b>	<b>21</b>
7.1	Stručný popis souborů . . . . .	21
7.2	Hlavní část . . . . .	21
7.2.1	Použité knihovny . . . . .	21
7.2.2	E-ink displej . . . . .	21
7.2.3	Sloty . . . . .	22
7.2.4	Konečný automat . . . . .	23
7.2.5	Přerušení . . . . .	24
7.2.6	Spuštění vedlejší části . . . . .	25
7.2.7	IR senzor . . . . .	26
7.3	Vedlejší část . . . . .	26
7.3.1	APA102 . . . . .	26
7.3.2	WS281X . . . . .	27
<b>8</b>	<b>Realizace hardwaru</b>	<b>29</b>
8.1	Nákup komponent . . . . .	30
8.2	Raspberry Pi Zero W . . . . .	30
8.3	Zdroje . . . . .	30
8.3.1	Maximální počet LED . . . . .	30
8.4	LED pásy . . . . .	31
8.5	Senzory . . . . .	31
8.5.1	Modul světelného senzoru . . . . .	31
8.5.2	Modul zvukového senzoru . . . . .	32
8.5.3	TSOP31238 . . . . .	32
8.6	Sloty . . . . .	32
8.7	Display . . . . .	33
8.8	Relé . . . . .	33
8.9	Úrovňové měniče . . . . .	33
8.10	Pojistka . . . . .	33
8.11	Skříň SP100 . . . . .	33
8.12	Ostatní . . . . .	34
<b>9</b>	<b>Testování</b>	<b>35</b>
9.1	První/Testovací realizace . . . . .	35
9.1.1	Hlavní zjištění . . . . .	35
9.2	Vývoj a ladění . . . . .	36
9.3	Uživatelské testování . . . . .	36
9.3.1	Spuštění programu . . . . .	36
9.3.2	Přidání pásku . . . . .	37
9.3.3	Ovládání pomocí tlačítek . . . . .	37
9.3.4	Ovládání pomocí senzoru . . . . .	38
9.4	Zátěžové testy . . . . .	38
<b>10</b>	<b>Závěr</b>	<b>39</b>
<b>A</b>	<b>Instalační příručka</b>	<b>41</b>
A.1	Raspberry Pi OS . . . . .	41
A.2	Dawn . . . . .	42
<b>B</b>	<b>Obrázky komponent</b>	<b>43</b>
	<b>Obsah přiloženého média</b>	<b>51</b>

## Seznam obrázků

3.1	MINI 21 key remote . . . . .	6
3.2	SP108E . . . . .	6
3.3	SP107E . . . . .	7
3.4	K-1000C . . . . .	7
5.1	Porovnání spotřeby RPi Zero W a Zero 2 . . . . .	12
5.2	Komunikační protokol WS281x . . . . .	14
5.3	Datový cyklus APA102 . . . . .	14
6.1	Naivní návrh hardwaru . . . . .	17
6.2	Naivní návrh softwaru . . . . .	18
6.3	Naivní návrh konečného automatu typu Moore . . . . .	18
6.4	Návrh zapojení s jedním slotem . . . . .	20
7.1	Příklad stromové struktury menu . . . . .	22
B.1	Měnič napětí a SP100 . . . . .	43
B.2	Pojistky . . . . .	43
B.3	IR přijímač a modul relé . . . . .	44
B.4	Světelný a zvukový senzor . . . . .	44
B.5	Displej a expander . . . . .	44
B.6	Konektor JST a GX16 . . . . .	45

## Seznam tabulek

3.1	Vlastnosti Mini 21 . . . . .	5
3.2	Vlastnosti SP1008E . . . . .	6
3.3	Vlastnosti SP107E . . . . .	7
3.4	Vlastnosti K-1000C . . . . .	7
4.1	Proudová zatížitelnost kabelů . . . . .	10
5.1	WS2812b parametry . . . . .	13
5.2	WS2815 parametry . . . . .	13
5.3	APA102 parametry . . . . .	13
8.1	Zapojení RPi headeru . . . . .	29

**Seznam výpisů kódu**

7.1	Struktura Menu . . . . .	22
7.2	Struktura Slot . . . . .	22
7.3	Ukázka implementace Moore . . . . .	23
7.4	Vlastní datový typ . . . . .	23
7.5	Příklad implementace stavu automatu . . . . .	24
7.6	Obsluha displeje . . . . .	25
7.7	Příklad spouštěcího řetězce . . . . .	26
7.8	Třída APA102 . . . . .	27
7.9	Využití před připravené třídy z colorschemes . . . . .	27
7.10	Příklad použití knihovny WS281X . . . . .	27



*Chtěl bych poděkovat především panu Ing. Pavlovi Kubalíkovi, Ph.D. Dále bych chtěl poděkovat svým rodičům, kteří mi umožnili vystudovat vysokou školu. Nakonec vzdávám čest dvěma jednodeskovým počítačům Raspberry Pi Zero W a Zero 2, které padly za oběť dopsání této práce.*

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 10. května 2022

.....

## Abstrakt

Tato práce se zabývá návrhem a implementací řídicí jednotky pro individuálně adresovatelné led pásky na platformě Raspberry Pi Zero. Hlavní využití pro tuto jednotku je osvětlení venkovních ploch (altánů, podezdívek střech, bazénů a podobné), interiérů budov a vnitřků automobilů. Jednotka podporuje ovládání z SSH konzole, E-ink displeje a senzorů. Pro displej byla v rámci práce vytvořena knihovna, implementující uživatelské rozhraní. Součástí práce je také text o vývoji a ladění na Raspberry Pi Zero. Na konci práce je uživatelské testování celé aplikace.

**Klíčová slova** RaspberryPi, kontroler, E-ink displej, LED pásek, WS281X, APA102

## Abstract

This work deals with the design and implementation of a control unit for individually addressable LED strips on the Raspberry Pi Zero platform. The main use for this unit is the lighting of outdoor areas (gazebos, roof bases, swimming pools, etc.), building interiors and car interiors. The unit supports control from SSH console, E-ink display and sensors. A library implementing user interface, was created for the display. The work also includes a text on the development and debugging of the Raspberry Pi Zero. At the end of the work is user testing of the entire application.

**Keywords** RaspberryPi, controler, E-ink display, LED strip, WS281X, APA102

## Seznam zkratek

CMOS	Complementary Metal Oxide Semiconductor
DC	Direct Current
DIY	Do It Yourself
DMX	Digital MultipleX
EPD	Electronic Paper Display
GND	GrouND
GPIO	General Purpose Input Output
HD	High Definition
I/O	Input/Output
IC	Integrated Controler
IPXX	Ingress Protection
IR	Infra Red
LED	Light Emiting Diode
LLS	Logic Level Shifter
OLED	Organic Light Emitting Diode
OS	Operating System
PWM	Pulse Width Modulation
RF	Radio Frequency
RGB	Red Green Blue
RPi	Raspberry Pi
SPI	Serial Peripheral Interface
SSH	Secure Shell
TTL	Transistor Transistor Logic
USB	Universal Serial Bus



## Kapitola 1

# Úvod

*Světelné diody (LED a OLED) opouštějí okrajové části trhu s indikátory a dostávají zelenou jako světelné zdroje budoucnosti. Umělé osvětlení ročně spolkně pětinu celosvětové spotřeby energie, v Evropě dokonce celou čtvrtinu. Světelné diody jsou v osvětlování alternativou, která tuto nepříznivou situaci může v budoucnu výrazně změnit, a tak přispět i ke zmírnění zátěže životního prostředí.\*)[1]*

Úkolem této práce je sestavit a naprogramovat řídicí jednotku individuálně adresovatelných LED na platformě Raspberry Pi. Hlavním využitím pro tuto jednotku je osvětlení venkovních ploch (altánů, podezdívek střech, bazénů a podobně), interiérů budov a vnitřků automobilů. Úkol má dvě části: návrh připojení periférií a programování jednodeskového počítače. Součástí první části budou informace, jak správně vybrat LED pásky a napájecí zdroje. Součástí druhé části bude, jak debugovat jednodeskový počítač Raspberry Pi.

Ovládací periférie jsou SSH, IR ovladač a fyzická tlačítka. K jednotce se lze připojit pomocí Wi-Fi a nebo USB. Konfigurace jednotky probíhá přes protokol SSH a linuxový terminál. Hlavní funkcí je přidání/odebrání LED pásku, nastavení senzoru a tlačítek. Samotné ovládání je hlavně skrze IR ovladač a tlačítka. K zobrazení informací o stavu bude sloužit e-ink display.

Součástí práce bude také příklad konfigurace/zapojení jednotky, LED pásku, zdroje a čidel. Toto téma jsem si zvolil se záměrem použít výsledný produkt k osvětlení rodinného bazénu a altánu na pozemku mých rodičů. Hlavním důvodem, proč jsem nezvolil komerční řešení, je, že žádné nevyhovovalo mým požadavkům.



# Popis problému, specifikace cíle

Mým cílem je vytvořit zapouzdřenou jednotku, ke které budu moci připojit různé individuálně adresovatelné LED pásy. Tuto jednotku bych chtěl použít pro osvětlení venkovního altánu a bazénu, popřípadě obývacího pokoje. Měla by tedy být jednoduše modifikovatelná podle potřeb projektu.

Jednotka musí být schopná najednou řídit až několik stovek RGB diod. Tyto diody často potřebují striktní časování pro správnou funkci a nezanedbatelné množství operační paměti pro uložení stavu. Bude proto potřeba prozkoumat hardwarové schopnosti platformy Raspberry Pi.

Dalším problémem bude samotné provozování pásků z elektrotechnického hlediska. Jejich několikametrová délka s sebou přináší svoje problémy, jako například úbytek napětí na vedení nebo velký příkon při maximálním jasu.

## 2.1 Shrnutí funkčních požadavků

- Možnost připojení dvou a více led pásků
- Podpora více typů led. Alespoň jeden 12V a 5V
- Jednotku bude možno ovládat pomocí tlačítek, dálkového ovládání a SSH
- (Pro obsluhu k jednotce připojte Waveshare 2.7 e-ink display a vytvořte vhodné grafické rozhraní) Tento typ displaye je nevhodný
- Jednotka bude obsahovat senzor zvuku a světla sloužící k automatickému spouštění





## Kapitola 3

# Existující řešení

Většina existujících a volně dostupných řešení na trhu pochází z Číny. Největší příkoří, se kterým jsem se potýkal, při průzkumu trhu, byla jazyková bariéra. Většina oficiální dokumentace, katalogů produktů a návodů je nedostatečně nebo špatně přeložená do angličtiny.

Zde jsem chtěl uvést jednotlivé firmy a jejich řešení. Nakonec jsem tak neučinil, protože většina firem má podobná nebo stejná řešení. Často se jednalo o stejné produkty, pouze s jinou značkou na obalu (např SP107E, který vyrábí firma Shen Zhen ALITOVE Electronic Technology Co., Ltd. a BTF-LIGHTING Technology Co., Ltd.). Proto zde uvedu několik druhů RGB kontrolérů, které jsou ve stejné kategorii jako moje řešení.

Všechna níže uvedená zařízení jsou schopná ovládat pouze jeden LED pásek. Některá ale umožňují synchronizaci nebo hromadnou zprávu. Jak toho dosahují, je popsáno u každého jednotlivě.

### 3.1 Mini 21 Key RF/IR Remote

Jedná se pravděpodobně o nejjednodušší typ led kontroléru na trhu. Ovládání i samotné nastavení, je možné pouze pomocí RF nebo IR dálkového ovladače. Tento model a jemu podobné jsou používány pro napájení a řízení jen několika desítek LED. Maximální počet LED uvedený výrobcem u tohoto modelu je 1024. K dosažení tohoto čísla je ale potřeba externí napájecí zdroj. [2]

Tento model se dá pořídit okolo 10\$ na portálu <https://www.amazon.com/>. Podpora různých typů LED je ale slabá. Je možné připojit pouze pásy WS201X 5V varianty.

■ **Tabulka 3.1** Vlastnosti Mini 21

Klady	Zápory
Velmi levné	Podporuje pouze 5V led pásy
	Nevhodné pro delší pásy
	Ovládání pouze pomocí dálkového ovladače
	Jeden ovladač ovládá všechny pásy stejně
	Nepodporuje vlastní režimy blikání



■ Obrázek 3.1 MINI 21 key remote [2]

## 3.2 SP1008E

Díky možnosti napájení 5 až 24V tento model umožňuje ovládat celou škálu chytrých LED pásků. Ovládání a nastavení probíhá přes mobilní aplikaci (Android i iOS) a síť Wi-Fi. Připojení probíhá buď v režimu AP (Kontroler si vytvoří svoji vlastní Wi-Fi síť.), nebo STA (Kontroler je připojen do stejné Wi-Fi sítě jako mobilní telefon). Při použití druhé možnosti je možnost připojit SP1008E do chytré domácnosti např. Google Home.[3]

Tento model se dá pořídit za 24\$ na portálu <https://www.amazon.com/>. Neumožňuje vlastní režimy blikání a je schopný ovládat okolo 2000 LED dlouhý pásek.

■ Tabulka 3.2 Vlastnosti SP1008E

Klady	Zápory
Integrace chytré domácnosti	Ovládání pouze pomocí sítě Wi-Fi
Podpora různého napětí LED pásků	Nutnost instalace mobilní aplikace
	Nepodporuje vlastní režimy blikání



■ Obrázek 3.2 SP108E [3]

## 3.3 SP107E

Vzhledově a funkčností je podobný předchozímu s několika podstatnými odlišnostmi. Ovládání probíhá přes rozhraní Bluetooth a mobilní aplikaci (Android i iOS). Hlavním tahákem tohoto výrobku je ovládání zvukem/hudbou. Toho je docíleno vestavěným mikrofonom nebo připojením 3,5mm audio jack konektoru.[4]

Tento model se dá pořídit za 22\$ na portálu <https://www.amazon.com/>. Umožňuje nastavení barvy pro 8 přednastavených vzorů blikání. Kromě bluetooth neumožňuje jiný rozumný způsob ovládání.

■ **Tabulka 3.3** Vlastnosti SP107E

Klady	Zápory
Možnost ovládání pomocí hudby	Nepodporuje vlastní režimy blikání
Podpora různého napětí LED pásků	Nutnost instalace mobilní aplikace



■ **Obrázek 3.3** SP107E [4]

### 3.4 K-1000C

Nejpokročilejší ze všech zmíněných modelů. Podporuje jak individuálně adresovatelné, tak jednoduché RGB pásky. Umožňuje řetězení sama se sebou a díky tomu podporuje teoreticky neomezené množství LED. Díky SD slotu a desktop aplikaci je možné programovat vlastní styly blikání. Podporuje také protokol DMX, který se používá v zábavním průmyslu k ovládání světelných show.[5]

Tento model se dá pořídit za 51\$ na portálu <https://www.amazon.com/>. Pro můj projekt je nevhodný, hlavně díky vysoké ceně a nepřehledné dokumentaci.

■ **Tabulka 3.4** Vlastnosti K-1000C

Klady	Zápory
Možnost nastavení vlastních režimů	Vysoké cena oproti konkurenci
Podpora různého napětí LED pásků	Nedostatečný návod k PC aplikaci
Podporuje všechny oblíbené druhy LED pásků	
Umožňuje řetězit jednotky	
Možnost ovládání pomocí standardu DMX	



■ **Obrázek 3.4** K-1000C [4]



## Kapitola 4

# Teorie

### 4.1 Ohmův zákon

„Ve vodiči, kterým prochází elektrický proud, lze měřením stanovit dvě veličiny, a to elektrický proud  $I$ , naměřený v ampérech, a elektrické napětí, naměřené ve voltech. (...) Veličina  $R$  vyjadřuje vlastnosti prostředí, kterým prochází elektrický proud, a nazýváme ji elektrickým odporem nebo také rezistancí vodiče. (...) Platí vztah:

$$R = \frac{U}{I}(\omega; V, A), \quad (4.1)$$

který se nazývá Ohmovým zákonem. “[6, kapitola 2.2.1]

### 4.2 Výkon elektrického pole

„Elektrická práce, kterou vykoná stejnosměrný proud mezi dvěma místy v proudovém obvodu za určitou dobu, je dána napětím  $U$  mezi těmito místy, proudem  $I$  a dobou  $t$ , po kterou proud obvodem prochází. (...) Práci vykonanou za jednotku času nazýváme výkonem  $P$  a platí vztah:

$$P = UI(W; V, A), \quad (4.2)$$

Jednotkou je watt (W). “[6, kapitola 2.5]

### 4.3 Úbytek napětí na vedení

Protože některé pásky mohou být i několik metrů dlouhé, dochází na nich k nezanedbatelnému úbytku napětí. „Průchodem proudu vedením vzniká úbytek napětí ve vedení  $\Delta U_V$ , pro který platí vztah:  $\Delta U_V = R_v I$ “ [7, kapitola 1.7]. Je tedy zřejmé, že pro zajištění správné funkce integrovaných obvodů LED budeme muset napětí po nějaké délce pásku obnovit.

### 4.4 Mooreův stroj

V informatice se za Mooreův stroj/automat označuje konečný automat s výstupem, jehož hodnota je závislá pouze na vnitřním stavu automatu. Vstup automatu tedy neovlivní okamžitý nebo aktuální výstup, ale až ten následující. Jeho protějškem je Mealyho stroj, jehož výstup je určen jak vnitřním stavem, tak vstupem.[8][9]

**Mooreův stroj** je definován jako šestice:

- konečný soubor stavů,
- počáteční stav,
- konečný soubor vstupní abecedy,
- konečný soubor výstupní abecedy,
- přechodová funkce mapující stav a vstupní abecedu na následující stav,
- výstupní funkce mapující stav na výstupní abecedu.

## 4.5 SSH

„SSH (Secure Shell) je v informatice označení pro program a zároveň pro zabezpečený komunikační protokol v počítačových sítích, které používají TCP/IP. SSH byl navržen jako náhrada za telnet a další nezabezpečené vzdálené shelly (rlogin, rsh apod.), které posílají heslo v nezabezpečené formě a umožňují tak jeho odposlechnutí při přenosu pomocí počítačové sítě. Šifrování přenášených dat, které SSH poskytuje, slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť, jako je například Internet.“ [10]

## 4.6 Pulzně šířková modulace

„Pulzně šířková modulace, neboli PWM (Pulse Width Modulation) je diskretní modulace pro přenos analogového signálu pomocí dvouhodnotového signálu. Jako dvouhodnotová veličina může být použito například napětí, proud nebo světelný tok. Signál je přenášen pomocí střídavy. Pro demodulaci takového signálu pak stačí dolnofrekvenční propust. Vzhledem ke svým vlastnostem je pulzně šířková modulace často využívána ve výkonové elektronice pro řízení velikosti napětí nebo proudu. Kombinace PWM modulátoru a dolnofrekvenční propusti bývá rovněž využívána jako levná náhrada D/A převodníku.“ [11]

## 4.7 Průřez vodiče

■ **Tabulka 4.1** Proudová zatížitelnost kabelů [12]

Průřez jádra [ $mm^2$ ]	Proud [ $A$ ]
Vodiče měděné	
0,5	11
0,75	15
1	17
1,5	23
2	26
2,5	30
4	41
Vodiče hliníkové	
2,5	21
4,5	28

## Kapitola 5

# Analýza

### 5.1 Výběr komponent

Ze zadání, od pana vedoucího, jsem měl pevně určeny některé komponenty. Jedná se o jednodeskový počítač Raspberry Pi Zero a Waveshare 2.7" e-Paper displej čepce pro Raspberry platformu.

**Výkonné** komponenty pro napájení slotu a logických částí:

- 12V zdroj,
- 5V zdroj,
- 2 sady po 4 relé,
- LED pásky různých druhů,
- logický úrovněový měnič 3,3-5V;

**Logické** komponenty pro funkčnost řadiče:

- Raspberry Pi,
- e-ink display,
- prahové čidlo světla a zvuku,
- IR senzor;

**Ostatní** komponenty:

- konektory pro napájení,
- vypínač na 230V,
- pojistka,
- skříňka;

#### 5.1.1 Raspberry Pi

##### 5.1.1.1 Zero W

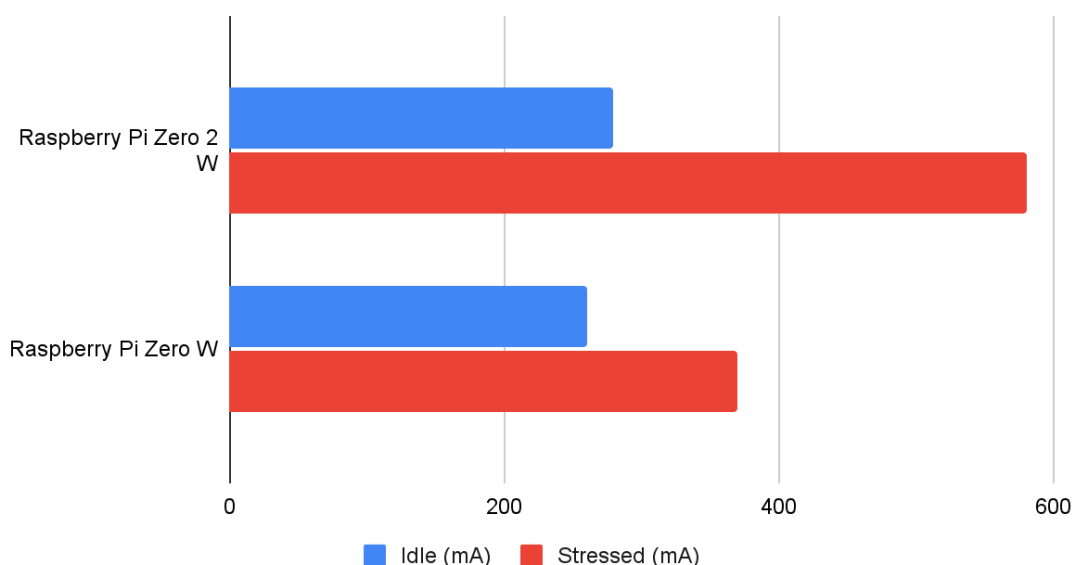
Extra levná a zmenšená varianta populárního Raspberry Pi. Zero W má navíc vestavěnou WiFi a Bluetooth. Zero je nejlevnější, nejmenší a nejspornější z minipočítačů Raspberry Pi. Hodí se k prototypování a do nenáročných vestavných a přenosných projektů. Zero je postaveno na starším čipsetu BCM2835 s jednojádrovým procesorem ARMv6, který ale stále nabízí dostatek výkonu i pro plynulé přehrávání Full HD videa.[13]

### 5.1.1.2 Zero 2

Srdcem Raspberry Pi Zero je RP3A0, na zakázku vyrobený zapouzdřený systém, navržený firmou Raspberry Pi v Anglii. S 4jádrovým 64-bitovým ARM Cortex-A53 procesorem, taktovaným na 1GHz, a 512MB SDRAM, je Zero 2x až 5x rychlejší než původní Raspberry Pi Zero. Stíněný radič pro bezdrátovou LAN poskytuje větší flexibilitu při práci s Raspberry Pi Zero 2 W.[14, Překlad]

### 5.1.1.3 Spotřeba

#### Raspberry Pi Zero 2 W Power Consumption



■ **Obrázek 5.1** Porovnání spotřeby RPi Zero W a Zero 2 [15]

## 5.1.2 LED pásky

Na trhu můžeme najít obrovské množství LED pásků. V této práci se ale budeme zabývat pouze pásky, které jsou takzvaně individuálně adresovatelné. To znamená, že každý pixel (dioda) se dá nastavit libovolně bez závislosti na předchozích nebo nadcházejících. Tyto pásy jsou podstatně dražší než jednodušší typy, a proto se kvůli omezeným finančním zdrojům budeme zabývat jen několika typy.

### 5.1.2.1 WS2812b

Jedná se o inteligentní LED světlo, které integruje ovládací obvod s RGB čipem do jednoho 5050<sup>1</sup> pouzdra. To zahrnuje interní digitální datový latch potr a tvarující zesilovací obvod pro kaskádové využití. Součástí je také precizní interní oscilátor a 12voltový napětím programovatelný stabilizátor proudu, zaručující konzistentní barevné parametry pixelu. Pouzdro má 4 piny: VDD, napájení ze zdroje, DOUT, datový výstup, VSS, zem a DIN, datový vstup.[16]

<sup>1</sup>5,0mm na 5,0mm



Pro přenášení dat používá proprietární protokol založený na pulzně šířkové modulaci. Pracuje s rámci o velikosti 24 b. Rámec se skládá z trojice bajtů, které reprezentují jednotlivé barvy. Prvních osm je modrá, pak červená a nakonec zelená. Pro ovládání tří sériově zapojených LED potřebujeme tedy tři rámce. První LED přečte první rámec a všechny následující vytvaruje, zesílí a pošle následující LED. Pro více informací nahlédněte na obrázek 5.2.

■ **Tabulka 5.1** WS2812b parametry

Parametr	Hodnota	Jednotka
Napájecí napětí	5	V
Logická jedna	$\geq 3,5$	V
Logická nula	$\leq 1,5$	V
Max. odběr proudu	Neuvedeno	mA
Max. příkon	Neuvedeno	W

### 5.1.2.2 WS2815

Je ze stejné rodiny čipů jako WS2812b s několika podstatnými změnami. Stejně jako předchozí disponuje i tento čip digitálním lachem a tvarovacím zesilovačem. Přítomný je již zmíněný precizní interní oscilátor a stabilizátor proudu. Novinkou je jeden záložní pin pro případ selhání jednoho pixelu v řadě. Pouzdro má šest (jeden nezapojený) pinů: VDD, napájení ze zdroje, DOUT, datový výstup, VSS, zem, DIN, datový vstup, BIN, záložní datový vstup a nezapojený pin VCC. [17]

Tento typ používá stejný protokol jako předchozí čip 5.1.2.1.

■ **Tabulka 5.2** WS2815 parametry

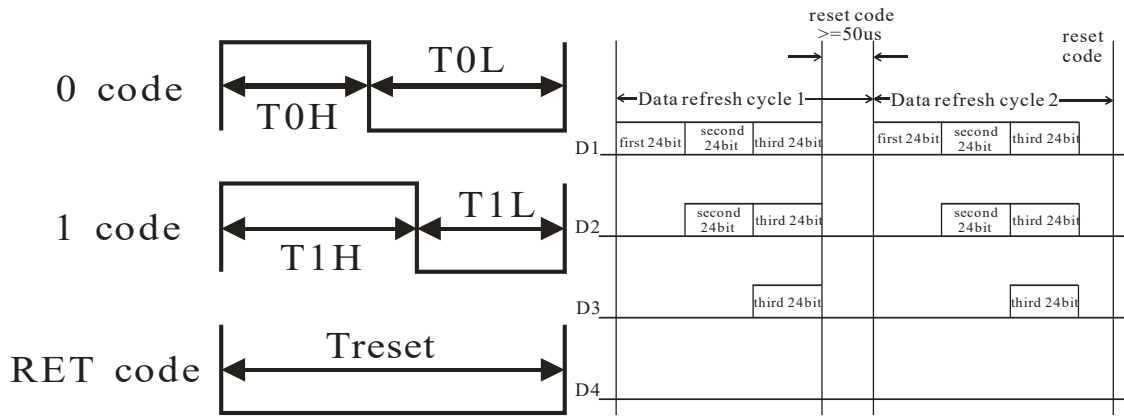
Parametr	Hodnota	Jednotka
Napájecí napětí	12	V
Logická jedna	$\geq 3,5$	V
Logická nula	$\leq 1,5$	V
Max. odběr proudu	Neuvedeno	mA
Max. příkon	Neuvedeno	W

### 5.1.2.3 APA102

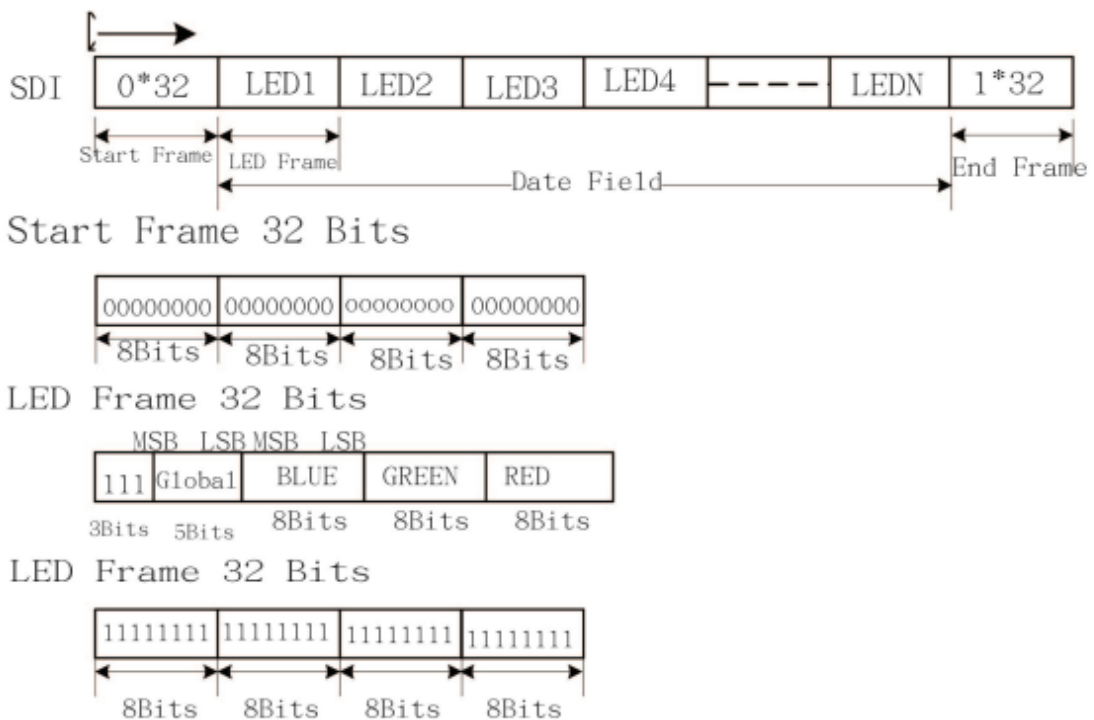
APA102, také zvané Super LED, jsou individuálně adresovatelné RGB LED diody. Podporují úpravu barev pomocí 256 stupňů šedi a 32 stupňů jasu. Ovládání probíhá pomocí datového a hodinového vstupu. Data se dělí na 32bitové rámce. První se odešle startovací rámec ( $0*32$ ), následují rámce pro jednotlivé pixely a jako poslední koncový rámec ( $1*32$ ). Rámec pixelu začíná třemi jedničkami, poté následuje 5bitová hodnota pro úroveň jasu a nakonec trojice bajtů popořadě reprezentující jasnost modré, zelené a červené. Pro více informací prozkoumejte obrázek 5.3. [18]

■ **Tabulka 5.3** APA102 parametry

Parametr	Hodnota	Jednotka
Napájecí napětí	5	V
Logická jedna	3,5-5,8	V
Logická nula	0-1,5	V
Max. odběr proudu	26,5	mA/barva
Max. příkon	128	mW/barva



■ Obrázek 5.2 Vlevo: PWM sekvence WS281x; Vpravo: Datový cyklus WS281x. [16]



■ Obrázek 5.3 Datový cyklus APA102 [18]

### 5.1.3 Napájecí zdroje

Pro potřeby tohoto projektu jsem se při výběru zdroje soustředil na dva hlavní parametry: cena a výkon. Kvůli nízkému napětí, které je požadováno elektronikou mého projektu, a omezenému rozpočtu jsem se rozhodl pro spínané zdroje. Protože budu potřebovat různé úrovně napájecího napětí, rozhodl jsem se pro nákup několika zdrojů elektrického napětí. Existují i spínané zdroje s dvěma úrovněmi napájecího napětí, ale ty jsou podle mé zkušenosti dražší nebo méně výkonné.

Protože z datasheetu pro některé LED, které budu používat, nebylo zřejmé jaký příkon mají, poohlédl jsem se na internetu po sekundárních zdrojích. Podle [19] je napětí na jedné diodě, při zobrazení bílé barvy, 0.013A/LED pro WS2815 a podle [20] 0.06A/LED pro WS2512b. Výkon určíme podle  $P = UI$  [6, kapitola 2.5].

## 5.2 Programovací jazyk a prostředí

Pro hlavní část práce jsem si zvolil jazyk C. Díky výběru displeje jsem měl od výrobce dvě možnosti: C nebo Python. Po dohodě s vedoucím jsem si zvolil C. Hlavní výhodou tohoto výběru je zrychlení programu. V předběžném testování displeje byla knihovna pro obsluhu přibližně o 50 % pomalejší. Také moje znalosti jazyka C jsou podstatně lepší než jeho opozice. Pro samotné ovládání LED pásku bude použit Python. Hlavním důvodem je kvalita a jednoduchost knihoven pro jejich obsluhu. [21]

Programovací prostředí jsem si zvolil Microsoft Visual Studio Code. Podpora nepřeborného množství rozšíření je ideální pro tuto úlohu.

### 5.2.1 Debug

Pro odlad'ování C'ěčkov'ého k'ódu na Raspberry Pi mám tyto možnosti (Použitelnost těchto řešení bude otestována v praktické části BP.):

- „printf(“Hello”);“,
- konzolový debugger např. gdb nebo Valgrind,
- vzdálené ladění pomocí křížového překladu v Microsoft Visual Studio.



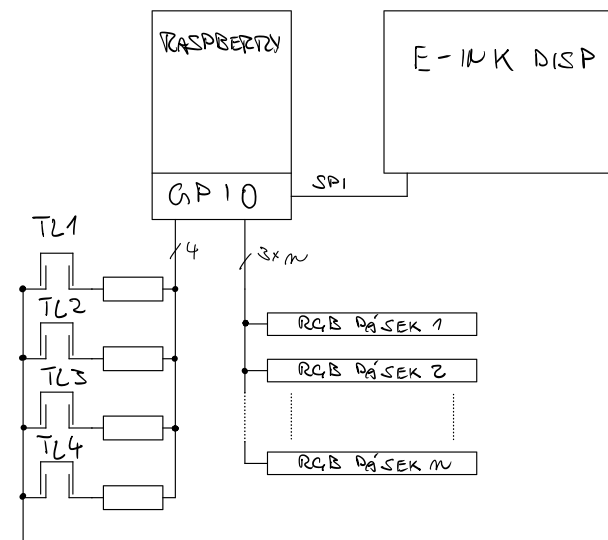
## Návrh řešení

## 6.1 Počáteční návrh

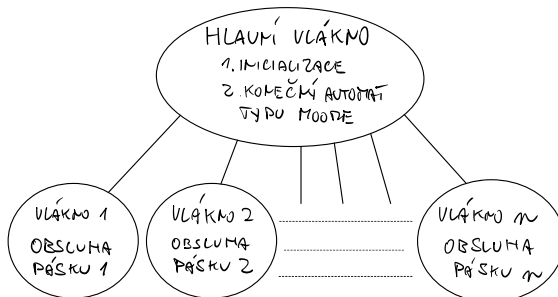
Už od začátku mi bylo jasné, že tento projekt bude mít dvě části. Hardwarovou a softwarovou. Po dohodě na výsledné podobě zadání vznikly dva diagramy reprezentující odpovídající části.

Hardwarová část (viz obrázek 6.1) zahrnuje I/O moduly, jako tlačítka a pásky, které jsou přímo připojené k jednodeskovému počítači. E-ink display je pak připojen pomocí linky SPI. Problémů tohoto návrhu je několik. Nemožnost připojit analogové senzory, špatná distribuce napájení pro led pásky a naivní myšlenka, že každý GPIO port Raspberry PI si je rovný.

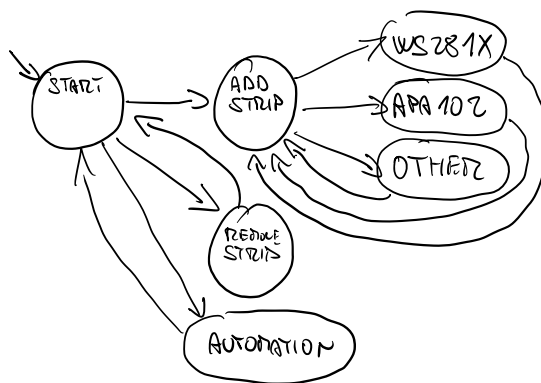
První softwarový návrh (viz obrázek 6.2) počítal s tím, že vše bude napsáno v jazyce C a to jako více vláknová aplikace. Srdcem programu je konečný automat typu Moore (viz obrázek 6.3). Jeho úkolem je zpracovávat přerušení a obsluhovat ostatní vlákna řídicí LED pásky. Od tohoto návrhu bylo upuštěno kvůli složité implementaci nových typů pásek.



■ Obrázek 6.1 Naivní návrh hardwaru



■ **Obrázek 6.2** Naivní návrh softwaru



■ **Obrázek 6.3** Naivní návrh konečného automatu typu Moore

## 6.2 Pokročilý/finální návrh

### 6.2.1 Software

Pro finální verzi návrhu jsem se rozhodl aplikaci rozdělit na dvě samostatné:

**hlavní** , napsaná v jazyce C a spravující vstupní a výstupní periferie:

- display,
- senzory,
- sloty;

**vedlejší** , napsaná v jazyce Python, která bude řídit LED pásy.

#### 6.2.1.1 Hlavní část

Pro zjednodušení přidělování zdrojů jsem se rozhodl do aplikace zavést tzv. sloty. Jedná se o reprezentaci fyzického konektoru na kontroléru a je tedy mezivrstvou mezi konektorem led pásku a GPIO jednodeskového počítače. Každý slot nemusí být nutně schopný připojení každého

typu LED pásku a to jaké zvládne, je dáno zdroji, které mu jsou přiděleny. Například jen některé sloty budou mít k dispozici PWM nebo SPI sběrnici.

Hlavní část aplikace zodpovídá za řízení kontroléru. Zpracovávání přerušení od senzoru a tlačítek, ovládání displeje, spouštění a správa vedlejších aplikací a přidělování zdrojů, to všechno budou úkoly této aplikace. Po provedení inicializace se spustí konečný automat typu Moore. Ten bude umožňovat nastavovat sloty a spouštět na nich automatizaci.

### 6.2.1.2 Vedlejší část

Soubor jednoduchých programů schopných spravovat právě jeden typ LED pásku. Tyto programy budou spouštěny z hlavní části, odkud také získají spouštěcí parametry. Těmi se bude nastavovat, jaký režim blikání bude zvolen. Spouštění a vypínání příslušného pásku bude docíleno spuštěním, respektive zastavením těchto programů z hlavní části. Ke každému slotu může být přidělen právě jeden takový program.

## 6.2.2 Hardware

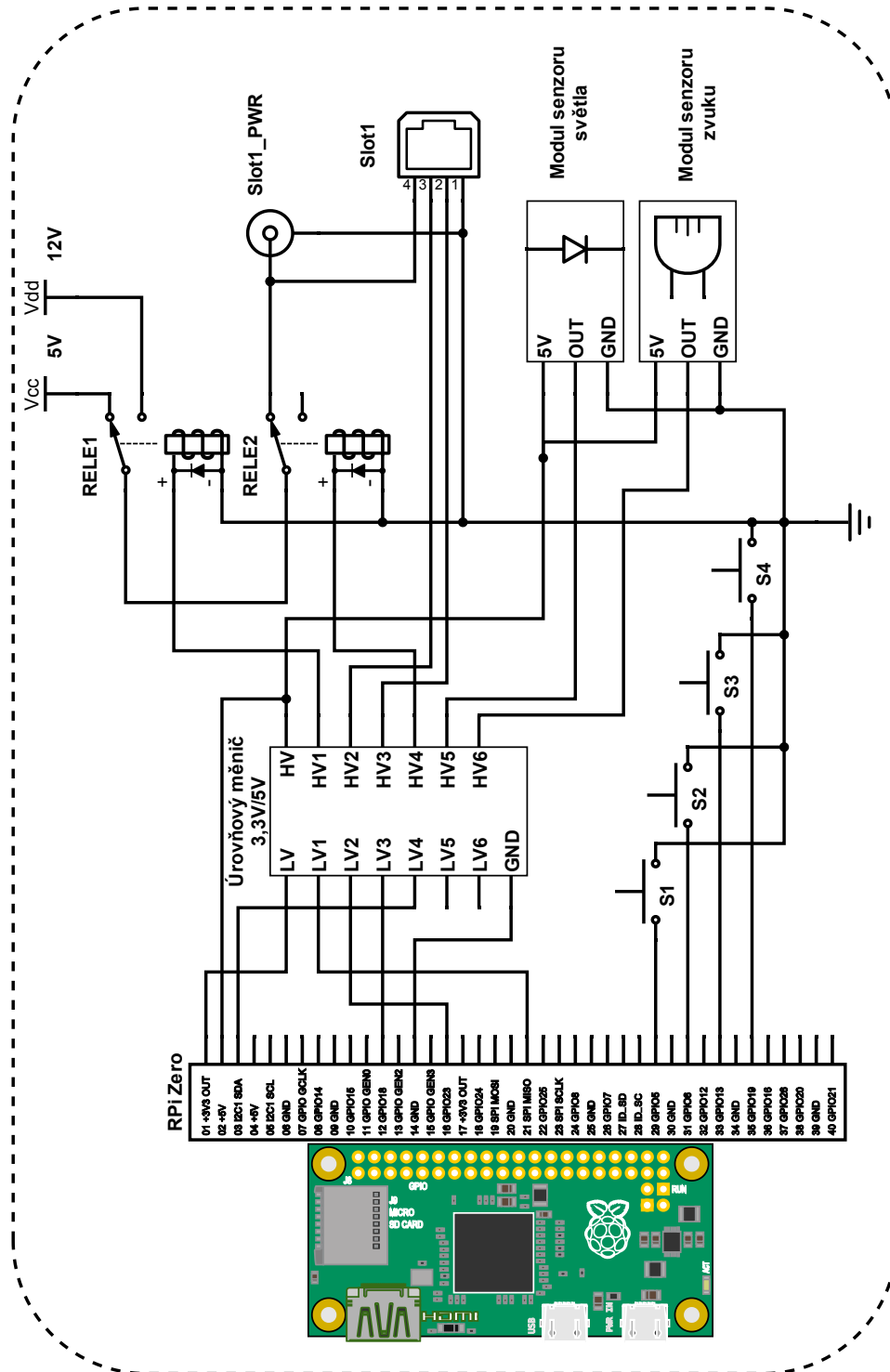
Tento návrh rozšiřuje počáteční návrh a přidává možnosti pro výše zmíněné sloty. Hlavním rozšířením je přidání dvou skupin přepínacích relé, které budou využity pro napájení pásku. První skupina relé bude přepínat, když se na slot dostane 5 V nebo 12 V. Druhá bude slot spínat (vypínat nebo zapínat napájení). Toto umožní ke každému slotu připojit pásky určené pro různá napětí.

Další novinkou je přidání úrovněového měniče z 3,3 V na 5 V a naopak. Ten bude sloužit jako mezimůstek mezi GPIO Raspberry PI (3,3 V), datovými vodiči RGB pásků (5 V), digitálními prahovými senzory (3,3 V nebo 5 V) a spínacím vstupem relé obvodů (5 V). Toto bylo přidáno k základnímu návrhu kvůli zaručení správnosti funkce připojených periférií.

Celá jednotka pak bude uzavřena do pojistkové skříně typu SP100 a bude umožněno odpojení všech připojených kabelů. Bude zde také hlavní vypínač, který umožní odpojit jednotku od napájení.

## 6.3 Uživatelská rozhraní

Uživatelská rozhraní budou: SSH konzole a E-ink display. Hlavním úkolem SSH konzole je nastavení kontroléru a připojených LED pásků. Rozhraní bude velmi jednoduché, nebude třeba zadávat nic jiného než čísla. Na E-ink displeji bude zobrazen stav zařízení a bude sloužit pro jednoduchou obsluhu pásku (zapnutí, vypnutí, změna režimu...).



■ Obrázek 6.4 Návrh zapojení s jedním slotem



# Realizace softwaru

## 7.1 Stručný popis souborů

- `./src/main.c` Toto je hlavní část mého programu.
- `./src/EPD.c` Obsahuje funkce pro obsluhu e-ink displeje.
- `./led/python/*` Tato složka obsahuje programy vedlejší části.

## 7.2 Hlavní část

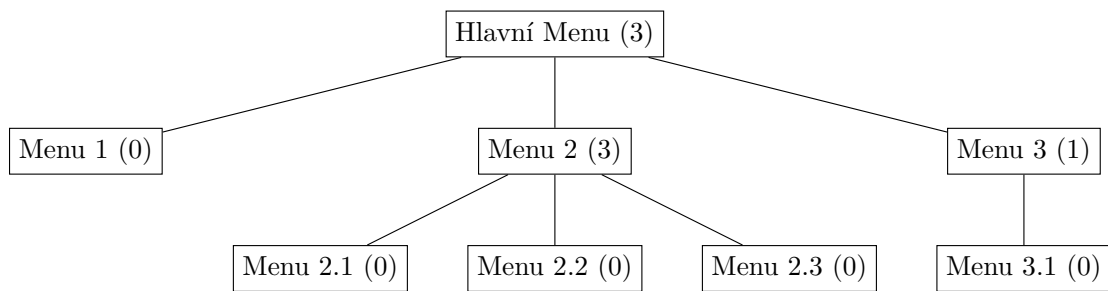
### 7.2.1 Použité knihovny

- `stdlib.h` pro práci s pamětí a systémová volání;
- `signal.h` pro zprávu programů spouštěných hlavní částí;
- `EPD.h` vlastní knihovna pro práci s E-ink displejem;
- `wiringPi.h` základní funkce pro práci s GPIO Raspberry Pi;
- `limits.h` QoL;

### 7.2.2 E-ink displej

Součástí zadání bylo vytvořit knihovnu pro ovládání Waveshare 2.7 e-ink display. Implementace tohoto je v souborech `./src/EPD.c` a `./src/EPD.h`. Zde najdeme strukturu `Menu` 7.1, která reprezentuje jednu položku v menu. Celé menu je potom reprezentováno jako n-nární strom. Každý uzel tohoto stromu je právě jedna položka menu. Součástí je také jednoduché uživatelské rozhraní.

Abychom mohli displej začít používat, musíme ho inicializovat, a to pomocí knihovny funkce `int EPD_2in7_init(int* dispSize, char** dispName)`. Parametry této funkce jsou použity pro inicializaci stromové struktury menu. `dispSize` je pole celých čísel určujících stromovou strukturu menu. Například při volání funkce s parametry `dispSize = {3,0,3,1,0,0,0,0}` bude mít kořen stromu 3 potomky. Jeho (zleva) první syn bude mít nula potomků, druhý bude mít tři a třetí jednoho potomka. Na třetí hladině budou už jen uzly bez potomků (pro představu nahlédněte zde 7.1). Jinak řečeno, číslo na indexu `i` určuje počet potomků uzlu na pozici `i` při



■ **Obrázek 7.1** Příklad stromové struktury menu

postupném číslování stromu od kořene po listy a to zleva doprava. Druhým parametrem funkce jsou názvy jednotlivých položek/uzlu.

Samotné ovládání displeje pak probíhá pomocí funkce `int EPD_2IN7_input(int input)`. Ta umožňuje posouvání výběru menu a zanořování nebo návrat ze submenu. Při volání dojde také k překreslení displeje, není tak nutné volat funkci `int EPD_2in7_draw_currMenu()`, která aktualizuje displej. Pokud však dojde ke změně jména či struktury položek menu, je vhodné ji zavolat. Tato knihovna obsahuje ještě několik dalších zajímavých funkcí, například pro skokové měnění vybrané položky, práci se stromovou strukturou a uvolnění alokovaných struktur.

Na čepci přípravku jsou přítomná čtyři tlačítka. Ta jsou použita pro jednoduchou navigaci displejem. První tlačítko posouvá výběr položky nahoru, druhé dolů, třetím se vstupuje do submenu a čtvrtým se z něho vrací. Tato tlačítka budou obsloužena pomocí přerušení (více informací 7.2.5.1).

■ **Výpis kódu 7.1** Struktura Menu

```

typedef struct Menu{

    void * parent;
    void ** children;
    int nChildren;
    char * name;
} Menu;
  
```

### 7.2.3 Sloty

Slot je struktura reprezentující jeden připojený pásek.

■ **Výpis kódu 7.2** Struktura Slot

```

typedef struct Slot{

    int pid; //program id of detached led program
    // {abspath or relative path to led program} {input variables}
    char program[256];
    LED_stripe_t stripT; //type of led strip installed

    int dma_chanel; //dma chanel used by led strip (-1 if not used)
    int led_count; //number of led on strip
    int led_brightness; //brightness of led strip in % [0-100]
    int led_mode; //selects mode of blinking for leds

    int sound_activated; //If 1 the strip will toggle on sound
    //0-> slot does not respond 1-> on during day 2-> on during night
    int light_activated;
  
```

```
int rele_on; //0->OFF; 1->ON
int rele_voltage; //0->5V; 1->12V

int rele_on_port;
int rele_voltage_port;

int data; //Data pin
int clock; //Clock pin
} Slot;
```

## 7.2.4 Konečný automat

Tato část programu je srdcem celé aplikace. Hlavní funkcí je nastavování slotů, které reprezentují připojené LED pásy. Konečný automat jsem se rozhodl implementovat pomocí nekonečné smyčky a podmíněného větvení 7.3. Stav automatu bude reprezentován vlastním datovým typem pomocí typedef enum.

■ **Výpis kódu 7.3** Ukázka implementace Moore

```
state current_state = start;
state next_state = start;

while(TRUE){

    switch(current_state){
        case start:
            ...
            next_state = set_strip;
            break;

        case set_strip:
            ...
            next_state = start;
            break;
    }
    current_state = next_state;
}
```

### 7.2.4.1 Konečný soubor stavů

Pro zpřehlednění implementace jsem si definoval vlastní datový typ reprezentující vnitřní stavy automatu. Nazývá se `state` a obsahuje všechny možné vnitřní stavy konečného automatu, jako první je uveden stav počáteční.

■ **Výpis kódu 7.4** Vlastní datový typ

```
typedef enum{

    start,
    add_strip,
    ...
    set_brightness,
    exit

} state;
```

### 7.2.4.2 Počáteční stav

Počátečním stavem je `start`.

### 7.2.4.3 Vstupní abeceda

Vstupní abecedou jsou čísla typu `int` v nějakém rozsahu. Ta jsou získána od uživatele skrze terminál a funkci `int getIntInput(int x, int y)`, kde `x` a `y` označují přijatelný rozsah vstupu.

### 7.2.4.4 Výstupní abeceda

Výstupním znakem mého automatu je funkce. Výstupní abecedou jsou tak všechny funkce, které můj automat bude volat.

### 7.2.4.5 Přejchodová funkce

Přejchodová funkce je realizovaná pomocí proměnných `current_state` a `ne_state` viz 7.5

### 7.2.4.6 Výstupní funkce

Je definovaná umístěním funkcí do `case state:`.

■ **Výpis kódu 7.5** Příklad implementace stavu automatu

```
case start:

    if(!DEBUG) system("clear");
    printf("Select one of the following:\n");
    for(int i = 0; i < slotsC; i++){
        printf("%d.", i + 1);
        printSlot(&slots[i]);
        printf("\n");
    }

    selectedSlot = getIntInput(1, slotsC) - 1;

    next_state = slot_control;

    break;
```

## 7.2.5 Přerušení

Realizované pomocí knihovny `wiringPi.h` a funkce `wiringPiISR` s argumenty `pin`, číslo GPIO, `mode`, hrana, na kterou bude přerušení reagovat, a `function`, obslužná funkce. Tohoto je využito pro obsluhu senzoru a tlačítek.

V rámci zvolené knihovní metody není možné nastavit kolikrát za jednotku času tato přerušení nastanou. Zámkity na vstupu tak způsobovaly několikanásobné spuštění obslužné metody při jednom chtěném vstupu. Toto bylo vyřešeno pomocí funkce `millis()`, která vrací počet milisekund od inicializace knihovny `wiringPi.h` pomocí funkce `wiringPiSetupGpio()`. Tato funkce byla přímo inspirovaná stejnojmennou funkcí z platformy Arduino (<https://www.arduino.cc/>).

### 7.2.5.1 Senzory

Při přerušení od prahových senzorů (světlo, zvuk), dojde v rámci obslužné metody k prozkoumání, zda byl nějaký slot nastaven na to, aby reagoval na světlo nebo zvuk. Pokud se zjistí,

že byl nastaven na zvuk (`sound_activated = SOUND_ON`), dojde k negaci `Slot.rele_on` a nastavení GPIO portu s číslem `rele_on_port`. Pokud byl nastaven na denní svícení (`light_activated = LIGHT_DAY`) nebo noční svícení (`light_activated = LIGHT_NIGHT`) dojde podle hrany vstupu k odpovídajícímu nastavení `Slot.rele_on` a odpovídajícímu GPIO.

Protože operace nastavení GPIO a porovnání několika číselných hodnot jsou operace časově nenáročné, je možné je všechny provést v rámci funkce pro obsluhu přerušení.

### 7.2.5.2 Tlačítka displeje

Tlačítka slouží pro navigování E-ink displeje 7.2.2. Protože přepsání displeje je časově náročná záležitost, byla tato zodpovědnost přesunuta na samostatné vlákno, vytvořené v inicializaci `init()` pomocí funkce z knihovny `wiringPi.h` `piThreadCreate`. Jediným vstupním argumentem této funkce je funkce, kterou bude vlákno vykonávat. Návratovou hodnotou je pak informace, zda se vytvoření vlákna zdařilo, či ne.

V samotné obsluze tlačítek je tak pouze zapsání do globální proměnné `int key_pressed` a otevření zámku `DISP_REFRESH_LOCK` pomocí funkce `void piUnlock(int lock)`. Tento zámek slouží pro uspaní vlákna, které jsme vytvořili výše. Zmáčknutí tlačítka tak umožní tomuto vláknu se probudit z funkce `void piLock(int lock)` a přepsat displej podle zmáčknutého tlačítka. 7.6

#### ■ Výpis kódu 7.6 Obsluha displeje

```
void INT_KEY_CONFIRM(){
    key_pressed = KEY_CONFIRM;
    piUnlock (DISP_REFRESH_LOCK);
    return;
}

PI_THREAD (dispTread){
    EPD_2in7_draw_currMenu();

    for(;;){
        piLock(DISP_REFRESH_LOCK);
        piLock(DISP_SHOW_LOCK);

        resolve_key_press();
        piUnlock(DISP_SHOW_LOCK);
    }

    return 0;
}
```

## 7.2.6 Spuštění vedlejší části

Spouštění vedlejší části probíhá v rámci konečného automatu 7.2.4. Využívá se systémových volání pomocí funkce `int System(char*)`. Celý tento proces zastřešuje funkce `int startSlot` a k ní opačná `int stopSlot` (argumentem těchto funkcí je ukazatel na `Slot`). Tyto funkce zodpovídají za korektní volání UNIXových příkazů pro spouštění a zastavování vedlejší části.

Po nastavení parametrů slotu (typ pásky, počet LED, programu, svítivosti) pomocí konečného automatu, dojde k zavolání funkce `startSlot`. Ta zkontroluje, zda jsou nastavené hodnoty validní, vytvoří spouštěcí řetězec (`slot.program` 7.7), pomocí systémového volání spustí odpovídající `*.py` program a uloží PID spuštěné aplikace. Tato část obsahuje trochu linuxové

magie, protože, návratová hodnota funkce `int System(char*)` není rovná návratové hodnotě spuštěného programu. Součástí spouštěcího řetězce je tak příkaz `echo $! > pid`, který nám uloží číslo PID do dočasného souboru. Z toho se dá jednoduše získat pomocí funkcí pro práci se souborovými streamy `fopen` a `fscanf`.

■ **Výpis kódu 7.7** Příklad spouštěcího řetězce

```
//LED_COUNT, DATA_PIN, CLOCK_PIN, LED_BRIGHTNESS, LED_MODE
char * apa102ExecString =
    "python3.9./led/python/APA102.py %d %d %d %d %d & echo $! > pid";
```

## 7.2.7 IR senzor

Pro obsluhu infra červené diody a přijímače jsem si zvolil knihovnu `lirc`. Z nedostatku času ale nebyla implementována jako součást konečného řešení.

## 7.3 Vedlejší část

Vedlejší část je soubor programů používaných pro řízení LED pásků připojených k jednotce. Nacházejí se ve složce `./led/python`. Tyto programy využívají knihoven, které jsem našel na internetu, a jejichž autoři jsou uvedeni u jednotlivých variant.

Každý z těchto programů má podobné volací argumenty. Jde o pětici čísel, která obsahují informace nutné k obsluze pásku. Prvním je počet LED umístěných na pásku. Tento parametr musí mít kladnou hodnotu (záporná čísla nedávají smysl), ale nutně nemusí reprezentovat reálné číslo LED. Může být menší, v tomto případě bude fungovat jen část pásku, nebo větší, což se dá použít pro testování paměťové náročnosti.

Druhým parametrem je číslo GPIO, který bude posílat data. V případě pásků vyžadujících striktní časování tento port musí podporovat PWM. Třetí parametr je závislý na typu LED. Pro pásky využívající PWM je toto číslo DMA kanál. Pro pásky vyžadující externí hodinový signál je to GPIO s hodinovým signálem.

Čtvrtý a pátý parametr určují svítivost LED v % a režim blikání. V základu je pro uživatele připraveno pět režimů pro vyzkoušení. Jednoduchou úpravou pytnových programů je možné tyto režimy změnit. Tyto programy lze při správném nastavení vstupních argumentů použít samostatně, bez spouštění hlavní části, například pro testovací účely nebo při implementaci vlastních režimů.

### 7.3.1 APA102

Program pro obsluhu LED pásků s diodami typu APA102. Tato implementace byla navržena a otestována na revizi C<sup>1</sup>, měla by ale fungovat i pro originální. Z analýzy 5.1.2.3 jsme se dozvěděli, že tento model využívá dvou vodičového komunikačního protokolu. Jeden vodič pro data a jeden pro hodinový signál. Díky tomu nepotřebujeme přesné časování a můžeme použít běžné GPIO porty.

Knihovna pohánějící tento program má dva přístupy k ovládání pásků. Jeden je přímý a druhý pomocí cyklů. První z nich umožňuje přímo ovládat barvu a jas jednotlivých pixelů pásků pomocí třídy `APA102`. Součástí této třídy jsou metody pro nastavování LED pásků (například `set_pixel`, `show`, `clear_strip`) a pomocné metody (`combine_color` a `wheel`). Můj kód těchto metod využívá nastavování jednoduchých režimů svícení. Příklad 7.8.

<sup>1</sup>APA102C

### ■ Výpis kódu 7.8 Třída APA102

```
from apa102_pi.driver import apa102

strip = apa102.APA102(num_led=LED_COUNT, bus_method='bitbang',
                      mosi=DATA_PIN, sclk=CLOCK_PIN, global_brightness=31)

for x in range(60):
    strip.set_pixel_rgb(x, 0xFFFFFF, bright_percent=LED_BRIGHTNESS)
strip.show()
```

Druhý přístup je pomocí tříd ze souboru `colorschemes.py`. Každá zde definovaná třída reprezentuje jeden režim svícení. Jsou to například: honička (`TheaterChase`), kolem dokola (`RoundAndRound`) a další. Tento přístup má výhodu v tom, že mi umožnil realizovat složitější režimy za velmi krátkou dobu. Příklad z mého kódu 7.9.

### ■ Výpis kódu 7.9 Využití před připravené třídy z `colorschemes`

```
from apa102_pi.colorschemes import colorschemes

my_cycle = colorschemes.StrandTest(num_led=LED_COUNT, pause_value=0,
                                   num_steps_per_cycle=LED_COUNT, num_cycles=-1, order='rgb',
                                   global_brightness=31, bus_method='bitbang', mosi=DATA_PIN,
                                   sclk=CLOCK_PIN)

my_cycle.start()
```

Tato implementace využívá knihovny `apa102-pi` ze stránky [www.github.com](http://www.github.com) od uživatele `tinue`, Martin [22]. Nachází se v souboru `./led/python/APA102.py`.

## 7.3.2 WS281X

Program pro obsluhu LED pásky s diodami typu WS281X. Protože LED pásky typu WS2815 a WS2812 používají stejný komunikační protokol 5.2, je možné je řídit pomocí stejného programu. Pro tento typ pásky je však nutné precizní časování, je tak možné ho řídit pouze PWM GPIO porty. RPi má 2 PWM jednotky, každá z nich má 2 kanály. Každý kanál je připojen na jeden GPIO. Bez komplikované hardwarové magie je tak možné řídit pouze čtyři pásky ve stejný čas.

Knihovna, kterou využívám, obsahuje třídu `PixelStrip` a strukturu `Color`. První z nich reprezentuje LED pásek, zprostředkovává inicializaci a umožňuje přímý přístup k nastavení jednotlivých pixelů. Druhá usnadňuje práci s barvami, které budou zobrazeny na pásku. Součástí repozitáře knihovny bylo několik podpůrných funkcí, které jsem využil ve svém programu (například `colorWipe`, `setColor`, `rainbow`). Tyto funkce pracují s třídou `PixelStrip` a reprezentují několik režimů blikání. Pro příklad nahlédněte na 7.10.

### ■ Výpis kódu 7.10 Příklad použití knihovny WS281X

```
# Create NeoPixel object with appropriate configuration.
strip = PixelStrip(LED_COUNT, LED_PIN, LED_FREQ_HZ,
                  LED_DMA, LED_INVERT, LED_BRIGHTNESS, LED_CHANNEL)
# Intialize the library.
strip.begin()

while True:
    theaterChase(strip, Color(127, 0, 0)) # Red theater chase
```

V mém projektu je tato knihovna využita pro řízení 5V pásky WS2812b a 12V WS2815. S několika úpravami by ale bylo možné řídit i ostatní pásky z této rodiny, a to i například

RGBW LED diody s označením SK6812RGBW. Kvůli nedostatku času podpora těchto diod nebude implementována.

Tato implementace využívá knihovny `rpi-ws281x-python` [23] ze stránky [www.github.com](http://www.github.com). Nachází se v souboru `./led/python/WS281X.py`.



# Realizace hardwaru

Když jsem začal s touto prací, měl nejméně zkušeností s výběrem hardwaru a prací s ním. Tak se stalo, že když jsem začal práci dávat dohromady, několik součástek odešlo do křemíkového nebe. Mezi ně patří jednodeskový počítač Raspberry Pi Zero 2, dvě nepájivá pole, 5V zdroj napětí a 2 metry LED pásku. Jejich oběť nebyla zbytečná, protože s každou chybou jsem se něco nového naučil, a to mi umožnilo vytvořit lepší produkt.

■ **Tabulka 8.1** Zapojení RPi headeru, tabulka odpovídá fyzické podobě headeru

GPIO	Zapojení	GPIO	Zapojení
3V3	Napájení LLS	5V	Napájení LLS
2	SLOT_1_RELE_ON	5V	Napájení RPi
3	SLOT_2_RELE_ON	GND	GND
4	SLOT_3_RELE_ON	14	SLOT_3_DATA
GND	GND	15	SLOT_3_CLOCK
17	EPD_RST	18	SLOT_1_DATA
27	TSOP31238_OUT	GND	GND
22	Odpojeno	23	SLOT_1_CLOCK
3V3	Odpojeno	24	EPD_BUSY
10	EPD_DIN	GND	GND
9	SLOT_1_RELE_V	25	EPD_DC
11	EPD_SCK	8	EPD_CS
GND	GND	7	SLOT_2_RELE_V
0	ID_SD	1	ID_SC
5	EPD_KEY1	GND	GND
6	EPD_KEY2	12	SLOT_3_RELE_V
13	EPD_KEY3	GND	GND
19	EPD_KEY4	16	LIGHT_SENSOR
26	SLOT_2_CLOCK	20	SOUND_SENSOR
26	GND	21	SLOT_2_DATA

## 8.1 Nákup komponent

Protože tuto práci jsem vytvářel po pandemii covidu-19 a během ruské invaze na Ukrajinu, byl můj výběr elektromateriálu značně omezený. Očekával jsem, že si vystačím s nákupem z jednoho internetového obchodu, nemohl jsem však být dál od pravdy.

Raspberry Pi Zero W a Zero 2 použité v tomto projektu bylo zakoupeno z e-shopu [www.rpishop.cz](http://www.rpishop.cz). Relé, zdroje, zesilovače a mechanický materiál (konektory, vodiče...) byl zakoupen z portálu [www.drateg.cz](http://www.drateg.cz). Led pásy z [www.aliexpress.com](http://www.aliexpress.com) a [www.digitalni-led.cz](http://www.digitalni-led.cz). Nástroje, které jsem potřeboval k pájení a montování, byly z webu [www.botland.cz](http://www.botland.cz).

## 8.2 Raspberry Pi Zero W

Tento přípravek, jak již bylo zmíněno, je pevně určen od vedoucího práce. V této sekci se budeme zabývat jeho zapojením v mé bakalářské práci. Při výběru zapojení jsem byl nejvíce omezen zapojením displeje. Protože je to čepcový modul, nedá se jeho zapojení nijak měnit. Připojil jsem ho tedy jako první. Následně jsem připojil senzory na GPIO, které neumí nic speciálního (PWM, datové sběrnice...). Poté jsem spočítal, kolik mi zbývá GPIO, a vydělil to 4 (počet pinu pro realizaci jednoho slotu) a zjistil jsem, že můžu vytvořit 3 sloty. Pro přesné zapojení nahlédněte do tabulky 8.1.

## 8.3 Zdroje

Po prozkoumání svých možností jsem zakoupil spínané zdroje od firmy SANPU. Oproti zdrojům konkurence, například firma MEAN WELL, byly značně levnější a navíc na ně byla sleva. Některé uživatelské recenze poukazyvaly na horší kvalitu použitých komponent. Já však budu používat okolo 60 % výkonu těchto zdrojů, takže by neměly být nijak významně namáhány.

Nakonec jsem se rozhodl pro koupi těchto dvou modelů:

- 12V 250W SANPU EPS250-H1V12 [24],
- 5V 36W SANPU PS36-W1V5 [25].

### 8.3.1 Maximální počet LED

Z analýzy víme, že spotřeba 5V pásky WS2812b je 60 mA/LED 5.1.3 a spotřeba RPi Zero 2 je 580 mA 5.1. Po převodu na výkon máme 300 mW/LED a 2.9 W. Protože hodlám zdroj využívat pouze na 60 % jeho uvedeného výkonu, mám k dispozici 21.6 W. A z toho už můžeme určit maximální počet LED na pásku.

$$(21.6 - 2.9) \div 0.300 \cong 60 \tag{8.1}$$

Maximální počet 5V LED, které můžeme bezpečně připojit k tomuto zdroji, je něco málo nad 60. To nám při standardní hustotě 60LED/m vystačí na pásek o délce jednoho metru. Je nutno poznamenat, že jsme si nechali poměrně značnou rezervu, a zdroj by tak mohl zvládnout více. K tomu by ale bylo nutné dostatečné chlazení a toho v uzavřené skříní nedosáhneme.

Stejný výpočet provedeme i pro 12V pásky WS2815. Máme tedy 13 mA/LED (156 mW/LED) spotřebu pro WS2815 5.1.3 a k dispozici máme 150 W výkonu ze zdroje. Tento zdroj je použit pouze pro napájení LED a jediné, co nás tedy zajímá, je spotřeba LED pásků.

$$150 \div 0.156 \cong 961 \tag{8.2}$$

Je tedy zřejmé, že tento zdroj je schopný napájet téměř 1000 LED typu WS2815. Opět je nutno zmínit 40 % rezervu, kterou jsme si nechali při výpočtu.

## 8.4 LED pásky

Pro testování byly zakoupeny tyto LED pásky:

- dvakrát jeden metr IP65 APA102C 60LED/m,
- jeden metr IP30 WS2812b 60LED/m,
- jeden metr IP30 WS2815 60LED/m,
- deset metrů IP30 WS2815 60LED/m,
- jeden metr IP65 SJ-10030-SK6812 30LED/m,

Pro to, jak jsou pásky připojeny ke kontroléru, nahlédněte do sekce 8.6.

## 8.5 Senzory

Protože Raspberry Pi postrádá převodník analogového signálu na signál digitální, mohl jsem si vybrat pouze prahové nebo inteligentní senzory. Protože všechny sběrnice již byly obsazeny ostatními periferiemi, vybral jsem si tyto tři senzory:

- modul světelného senzoru,
- modul zvukového senzoru,
- IR přijímač TSOP31238.

### 8.5.1 Modul světelného senzoru

Světelný senzor s nastavitelnou citlivostí a jednoduchým digitálním výstupem. Práh sepnutí se nastaví pomocí trimru umístěného na modulu. Zakoupeno z [www.rpishop.cz](http://www.rpishop.cz) [26]. Vybral jsem si tento modul, protože mi umožnil jednoduše detekovat, zda je den, nebo noc. Nepotřebuje být připojen k žádné sběrnici a stačí mu tedy pouze obyčejný digitální vstup. Pro zapojení nahlédněte do sekce 8.2.

#### Specifikace modulu

- Napájecí napětí: 3,3V-5V
- Logický proud: >15mA
- Komparátor: LM393
- Rozměry: 32mm x 14mm

## 8.5.2 Modul zvukového senzoru

Zvukový senzor s nastavitelnou citlivostí a jednoduchým digitálním výstupem. Práh sepnutí se nastaví pomocí trimru umístěného na modulu. Zakoupeno z [www.drateg.cz](http://www.drateg.cz) [27]. Vybral jsem si tento modul, protože mi umožnil jednoduše detekovat hlasité a krátké zvuky. Například tlesknutí, lusknutí nebo zaklepání na skříň. Nepotřebuje být připojen k žádné sběrnici a stačí mu tedy pouze obvyčejný digitální vstup. Pro zapojení nahlédněte do sekce 8.2.

### Specifikace modulu

- Velikost: 32mm\*17mm\*15mm (délka\*šířka\*výška)
- Čip: LM393, Elektretový kondenzátorový mikrofon
- Pracovní napětí: DC 4-6V
- Výstupní indikace: LED

## 8.5.3 TSOP31238

Hlavním důvodem, proč jsem si vybral tento produkt, je integrovaný předzesilovač. To mi umožní připojit senzor rovnou k GPIO RPi. Tento senzor je taky podporovaný v knihovně `lirc`. To mi značně usnadní práci při programování. Zakoupeno z [www.rpishop.cz](http://www.rpishop.cz) [28]. Pro zapojení nahlédněte do sekce 8.2.

### Specifikace modulu

- Detektor a předzesilovač v jednom balení
- Vnitřní filtr pro kmitočty PCM
- Lepší stínění proti rušení
- TTL a CMOS kompatibilita
- Napájecí napětí: 2,7V až 5,5V
- Lepší odolnost vůči okolnímu světlu

## 8.6 Sloty

Pro připojení pásky jsem si vybral konektor JST SM a to jeho 4pinovou verzi. Protože většina individuálně adresovatelných LED pásek má tento konektor už z továrny, byla tato volba nejrozumnější. Důležité je poznamenat, že některé pásky, a to i stejného typu, mohou mít jiné zapojení. Je proto vždy nutné ho zkontrolovat, než nový pásek připojíte k jednotce. Zapojení konektoru: první pin je připojen na GND, druhý je datový vodič, třetí je hodinový a čtvrtý je napájení. Pro některé pásky je 3. pin zbytečný, ponechte ho nezapojený, nebo uzemněný. [29]

Protože některé pásky dosahují až několika metrů, potřebujeme po určité délce obnovit napětí. Kvůli tomu jsem přidal extra napájecí konektory, které jsou určeny přesně k tomu. Bude na ně vždy přivedeno odpovídající napětí. To znamená, že pokud bude nastaven slot 1 na 5V pásek, bude na odpovídající napájecí konektor přivedeno 5 V. Pro konektor jsem se rozhodl vybrat 16mm GX16. Hlavním důvodem této volby byla dostupnost. Konektor je zapojen následovně: pin 1 je napájecí napětí a pin 2 je GND. [30]

## 8.7 Display

2,7palcový E-ink displej od společnosti WaveShare byl poskytnut vedoucím této práce, nemusel jsem si tak nic vybírat ani nakupovat. Tento modul displeje je čepec, který přesně sedí na header Raspberry Pi. Protože ale obsadí všechny piny sám pro sebe, zakoupil jsem si GPIO expander. Ten mi umožní připojit displej a zároveň používat ostatní GPIO, které by jinak byly nedostupné. Pro zapojení nahlédněte do sekce 8.2.

## 8.8 Relé

Úkolem relátek v mém projektu je spínání napájení slotů a výběr napětí napájení slotů. Protože RPi Zero má omezený proud, který se dá odebrat z jeho GPIO, hledal jsem řešení, které je možné napájet z externího zdroje. Také jsem potřeboval, aby relé přepínalo mezi dvěma výstupy. Našel jsem tento modul [31] obsahující čtyři relátka, který vyhovoval všem mým požadavkům. A protože jsem chtěl mít čtyři sloty (každý slot dvě relé), zakoupil jsem tyto moduly dva.

**Specifikace** modulu:

- 5V, 4-kanálové relé,
- každý kanál potřebuje 50-60mA proudu,
- relé: 49 VAC 5 A, 30 VDC 5 A,
- rozměry: 7 x 5,3 cm,
- LED indikace pro status výstupu.

## 8.9 Úrovňové měniče

Protože výstupní a vstupní napětí RPi GPIO je 3,3 V a všechny ostatní periferie vyžadují 5 V, musel jsem si pořídit měnič napětí. Našel jsem tento osmikanálový obousměrný měnič [32]. Jeho původní určení je pro mezistupeň mezi sběrnici I<sup>2</sup>C mikrokontroleru a zařízeními, k ní připojených. A to je přesně to, co potřebuji.

## 8.10 Pojistka

Pro výpočet pojistky potřebujeme zjistit proud, který jí bude protékat 4.14.2. Ten určíme tak, že sečteme výkon našich dvou zdrojů a vydělíme ho napětím sítě.

$$(250 + 36) \div 230 \cong 1.25 \quad (8.3)$$

## 8.11 Skříň SP100

Tuto skříň jsem vybral, protože je nehořlavá, dostatečně prostorná a nemusel jsem ji kupovat. Získal jsem ji od firmy EMSL s.r.o. [33] zdarma.

## 8.12 Ostatní

Pro propojení výkonových komponent byl použit kabel s měděným lanem CYH 2x0,75 [34]. Průřez  $0,75 \text{ mm}^2$  je kompromisem mezi úbytkem napětí na vodiči (sekce 4.3) a cenou. Kabel je také vyhovující z hlediska zatížení (tabulka 4.1). Pro zapínání a vzpínání byl zvolen kolébkový vypínač [35]. A pro možnost napájet RPi z 5V zdroje byl přidán páčkový přepínač [36].

### 9.1 První/Testovací realizace

Vytvořena pro otestování možností Raspberry Pi Zero W a Zero 2 W. Tato verze nepracovala se sloty, nýbrž umožňovala uživateli připojit pásky, jak chtěl. Při přidání pásku byl dotázán, jaký druh připojuje a následně si mohl rezervovat GPIO, které chtěl využít. Tato volnost ale přinesla spoustu potíží. Uživatel musel znát zapojení Raspberry Pi headeru a musel řešit připojení pásku přímo k Raspberry Pi. Stejný princip a potíže platily také pro vstupní periférie. Tyto obtíže vyřešila implementace slotu 7.2.3.

Hlavním přínosem této realizace bylo zjištění možností platformy Raspberry Pi, otestování E-ink displeje a softwarových možností ovládání LED pásku.

#### 9.1.1 Hlavní zjištění

- Waveshare EPD 2.7 E-ink display pro Raspberry Pi je prakticky nepoužitelný pro rozumnou konfiguraci. Hlavním důvodem je prepisovací rychlost displeje, která byla  $\sim 2$  s. Nastavení jednoho LED pásku by pak mohlo uživateli zabrat desítky sekund. Displej tak bude použit pro zobrazení nastavení slotů, přepínání jejich režimu a vypínání.
- Řadič bude moci ovládat pouze 2 pásky, ovládané pomocí pulzně šířkové modulace. Raspberry Pi Zero W a Zero 2 W mají 4 GPIO, které umožňují PWM a to 12, 13, 18, 19. 13 a 19. Jsou ale obsazeny tlačítky umístěnými na čepci E-ink displeje a není tak možné je realokovat jinam.
- Pásky vyžadující externí hodinový signál, např. APA102 jsou extrémně zatěžující na procesorový čas. Při spuštění na RPi Zero W zabere obsluha takového pásku až 80 % procesorového času, a to i přesto, že prepisovací rychlost se pohybuje okolo 2 snímků za sekundu.
- Maximální počet slotů bude 3. Protože každý slot zabere 4 GPIO (data, hodiny, relé pro výběr napětí a relé pro zapnutí) a celkový počet volných GPIO je 13 (27 využitelných - 3 senzory - 7 display - 4 tlačítka).
- Pro ovládání periférií (relé, senzory, pásky) bude nutný obousměrný převod logické úrovně z 3,3 V na 5 V. Nižší napětí způsobovalo nežádané chování 1. LEDky na pásku (následující fungovaly korektně díky internímu zesilovači v předcházejících diodách) a pro delší datové vodiče pásek přestal fungovat. Prahové senzory a relé moduly určené na 5 V nefungovaly vůbec.
- Raspberry Pi Zero 2 W je pro tuto úlohu značně vhodnější než původní Zero W. Hlavním důvodem jsou 4 vlákna procesoru novějšího modelu oproti 1 vláknu staršímu. 3 vlákna navíc

umožní konzistentní blikání LED při zatížení procesoru, např. při přepisování displeje zvýší maximální počet snímků za sekundu pásků ovládaných externími hodinami a celkovělepší uživatelský zážitek při ovládání přípravku.

- Pro ovládání LED pásků bude použit samostatný program v jazyce Python a to z důvodů lepší přehlednosti kódu a větší kvality knihoven pro obsluhu těchto pásků.
- Pro ovládání displeje bude využito jazyka C oproti jazyku Python. Hlavním důvodem je rychlost přepisování, která pro jazyk C byla několikanásobně větší.

## 9.2 Vývoj a ladění

Prvním krokem při vytváření tohoto projektu byl výběr programovacího prostředí. Protože jsem měl v plánu programovat na svém osobním počítači, muselo být vybrané prostředí schopné vyvíjet kód pro vzdálené prostředí. Chtěl jsem také, aby byl schopný podporovat více jazyků najednou, pro moji práci C a Python. Další nutností je možnost kompilovat kód přímo v prostředí. Po prozkoumání možností jsem došel k závěru, že si budu muset vybrat z následujících tří:

- SSH + NANO (nebo obdobný terminálový editor),
- Microsoft Visual Studio,
- Microsoft Visual Studio Code.

První možnost je nejjednodušší a nejsnadnější na nastavení. Protože povolení a připojení se k Raspberry Pi Zero pomocí SSH je součástí jeho počátečního nastavení, je tato možnost k dispozici jednoduše a bezbolestně. Je také nejméně náročná na systémové požadavky a díky terminálu umožňuje přímou kontrolu na RPi. Má však poměrně vysokou úroveň vstupních znalostí, potřebných pro efektivní práci. V případě editoru Nano je potřeba spousta nastavení například pro obarvování syntaxe, úpravu zarovnání, řádkování atd. Všechna tato nastavení se mění pomocí editace konfiguračních souborů a vyžadují tak jistou znalost ještě předtím, než napíšete první řádek vašeho kódu.

Microsoft Visual Studio je multifunkční vývojové prostředí od firmy Microsoft Corporation. Podle hlavní stránky umožňuje kódovat, sestavovat, ladit, nasazovat, spolupracovat, analyzovat a učit se. A je tomu opravdu tak. Toto IDE má nejvíce možností z celého našeho výběru. Jako jediné také umožňuje ladění za běhu programu. Při postupu pomocí návodu [37] se mi povedlo Raspberry Pi 4B připojit a vyzkoušet jednoduchý blikací projekt s jednou LED diodou a tlačítkem. Pro RPi Zero jsem už ale byl neúspěšný. Hlavním důvodem, proč toto řešení zklamalo, je 512 MB operační paměti, kterou najdeme na tomto jednodeskovém počítači. Při připojení MS VS se tato kapacita téměř ihned zaplní. Zero tak bylo velmi neresponzivní a při několika mých pokusech o „Hello World!“ projekt zamrzlo a vyžadovalo tvrdý restart.

## 9.3 Uživatelské testování

Hlavním úkolem uživatelského testování je odhalení softwarových chyb. Nejlépe se provádí ve větší skupině lidí různých znalostí. Díky tomu dojde k odhalení nejvíce nedostatků a chyb. Z časových důvodů budu tuto práci testovat sám.

### 9.3.1 Spuštění programu

Spustil jsem program pomocí:

```
./DawnLight
```



To způsobilo, že program byl neschopný spustit pytnové programy pro obsluhu LED. Respektive se při přidání pásku objevila chybová hláška: `cannot create pid: Permission denied`. Oprava: do manuálu bylo přidáno, aby uživatelé spouštěli program v administrátorském režimu.

Spustil jsem program pomocí:

```
sudo ./DawnLight
```

Zobrazilo se mi uživatelské menu:

```
Select one of the following:
1. LED strip NotUsed of 0 LEDs, in mode 0
2. LED strip NotUsed of 0 LEDs, in mode 0
3. LED strip NotUsed of 0 LEDs, in mode 0
```

### 9.3.2 Přidání pásku

Měl jsem zobrazené menu a program očekával uživatelský vstup a to v podobě čísla od 1 do 3. Zkusil jsem proto zadat písmeno. Program zobrazil hlášku: `Input is not a number` a čekal na můj další vstup. Vyzkoušel jsem proto ještě několik špatných vstupů (speciální znaky, čísla mimo rozsah, odstavec textu...). Setkal jsem se vždy s chybovou hláškou nesprávného vstupu.

Zadal jsem tedy 1 (Validní vstup) a na terminálu se mi zobrazilo toto menu:

```
Select one of the following:
-----Control-----
1. OFF
2. Mode: 0
3. No Light Control
4. No Sound Control
-----Settings-----
5. Not used
6. LED's: 0
7. Brightness: 0
```

Opět jsem zkusil všechny možné špatné vstupy, ale bezúspěšně. Vyzkoušel jsem proto všechny možnosti menu. Jako první jsem zkusil přidat pásek bez nastavení počtu ledek nebo svítivosti. Nic se však nestalo. Po nastavení počtu LED a zadání svítivosti se pásek rozsvítil.

Zkusil jsem přepínat ON/OFF, měnit mód svícení a ostatní parametry. Nic nechtěného se ale nestalo.

### 9.3.3 Ovládání pomocí tlačítek

Po přidání pásku se na displeji zobrazilo:

```
DawnLight
-----
>LED strip WS281X of 60 LEDs, in mode 1
LED strip is not used
LED strip is not used
-----
2022-05-05- 18:25:40
```

Zkusil jsem pomocí tlačítek vyjet z menu mimo displej. Ani nahoru ani dolů se nezdařilo a výběr položky se vždy zastavil. Zkusil jsem tedy tlačítka ON/OFF a změna modu. Obě fungovaly korektně.

## 9.3.4 Ovládání pomocí senzoru

### 9.3.4.1 Světlo

Po nastavení pásku do režimu `On during night` jsem vzal baterku a posvítíl na pásek. Ten se ihned vypnul. Po odebrání světla se zase vypnul. Zkusil jsem rychle zablikat pomocí mobilní aplikace (simulace blesku). Pásek se na chvíli zapnul a zase vypnul. Obdobně jsem otestoval i režim `On during day`. Ten fungoval totožně, pouze obráceně.

### 9.3.4.2 Zvuk

Nastavil jsem pásek do režimu `Toggle on sound`. Zkusil jsem tlesknout blízko senzoru zvuku. Nic se však nestalo. Oprava: Pomocí trimru na senzoru jsem zvýšil citlivost. Po této opravě už začal reagovat. Pásek se na tlesknutí, poklepání na kryt skříňky a lusknutí vypnul/zapnul.

## 9.4 Zátěžové testy

Z důvodu nedostatku času jsem zátěžové testy nedělal.



## Kapitola 10

# Závěr

V této práci se mi podařilo navrhnout a postavit Řídící jednotku pro individuálně adresovatelné LED na platformě Raspberry Pi. Ta umožňuje ovládat až 3 takové pásy různých druhů, využívajících různých napětí. Součástí jednotky jsou také senzory umožňující automatizaci ovládání. Výrobek jsem si částečně otestoval a poskytl jsem základní uživatelskou příručku. Tato práce umožní lidem zabývajícím se DIY a „baslením“ vyrobit si přesnou kopii mého řešení a slouží také pro získání informací ohledně individuálně adresovatelných pásek, jejich ovládání a napájení.



# Příloha A

## Instalační příručka

### A.1 Raspberry Pi OS

Prerevizity: micro SD karta o velikosti alespoň 8 GB, jednodeskový počítač Raspberry, napájecí kabel, notebook nebo stolní počítač.

1. Stáhnete si distribuci Raspberry Pi OS. Pro RPi Zero a Zero 2 doporučuji stáhnout variantu Lite.
2. Připravte si micro SD kartu a pomocí softwaru balenaEtcher nebo obdobnému nainstalujte operační systém. Po dokončení by se vám měl zobrazit disk s názvem boot.
3. Do tohoto disku/složky vložte prázdný soubor s názvem SSH. Je nutné, aby se jmenoval přesně takto a neměl žádnou příponu.
4. Vytvořte soubor se jménem wpa\_supplicant.conf. Ten slouží pro připojení k místní síti Wi-Fi. Alternativně můžete nastavit RPi jako USB gadget podle tohoto návodu.
5. Do tohoto souboru vložte

```
country=ISO_3166_alpha-2_STÁNÍ_KÓD
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="JMÉNO_SÍTĚ"
    psk="HESLO_SÍTĚ"
}
```

a upravte položku country, ssid a psk.

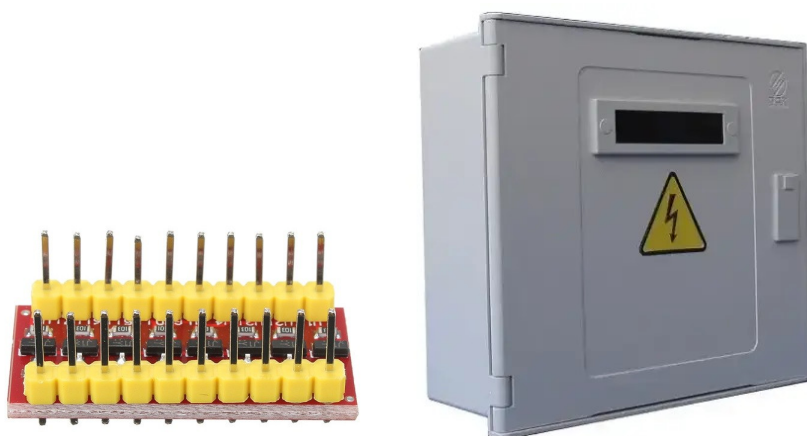
6. Vložte kartu do RPi a připojte napájecí napětí.
7. Počkejte, než RPi nabootuje (může trvat i několik minut) a zjistěte IP adresu, která byla RPi přidělena (buď pomocí IP snooping nástroje nebo nastavení DHCP serveru). Pokud se vám nedaří zařízení najít, zkontrolujte přihlašovací údaje.
8. Zapněte SSH klienta (doporučuji MobaXterm) a připojte se k SSH serveru RPi. Výchozí login je pi a výchozí heslo je raspberry.
9. Měl by se vám zobrazit terminál RPi.

## A.2 Dawn

Prerekvizitou k tomuto návodu je nainstalovaný a aktualizovaný Raspberry Pi OS a přístup k jeho terminálu.

1. Naklonujte si můj GitLab repozitář.
2. Přidejte práva pro správu GPIO
3. Nainstalujte WiringPi
4. Upravte definice v `main.c` podle potřeb vašeho programu
5. Skompilujte pomocí `make clean`
6. Spusťte s administrátorskými právy pomocí `sudo ./Dawn`

## Obrázky komponent



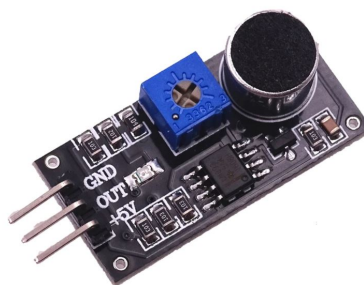
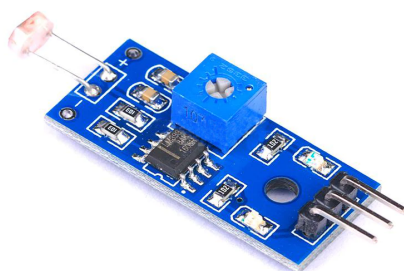
■ **Obrázek B.1** Vlevo: Osmikanálový měnič napětí [32]; Vlevo: SP100/NSP1P přípojková skříň [38].



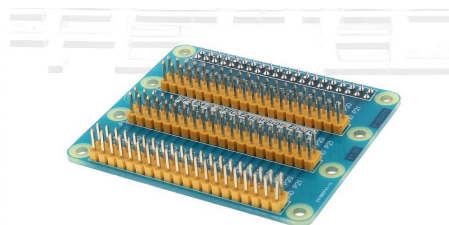
■ **Obrázek B.2** Vlevo: Trubičková pojistka [39]; Vpravo: Držák na pojistky [40].



■ **Obrázek B.3** Vlevo: IR přijímač TSOP31236 [28]; Vpravo: Relé 4 kanály 5V s optočlenem [31].

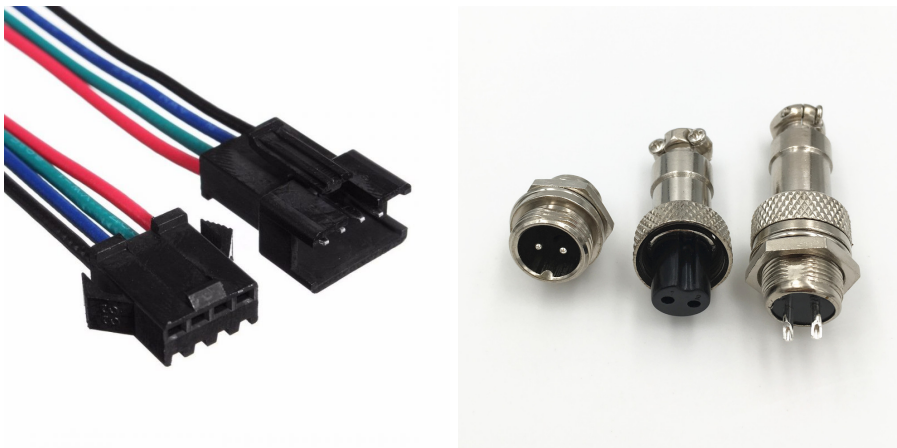


■ **Obrázek B.4** Vlevo: Modul světelného senzoru [26]; Vpravo: Modul zvukového senzoru [27].



■ **Obrázek B.5** Vlevo: E-ink displej čepce pro Raspberry Pi [41]; Vpravo: Expander GPIO. [42].





■ **Obrázek B.6** Vlevo: Konektor JST 2.54 SM 4P [29]; Vpravo: Konektor 16mm GX16 - 2 piny [30].



# Bibliografie

1. PUŽMANOVÁ, Rita. LED – světlo budoucnosti. *Časopis Světlo* [online]. 1998, roč. 2009, č. 5 [cit. 2022-04-20]. ISSN 1212-0812. Dostupné z: <http://www.odbornecasopisy.cz/svetlo/casopis/tema/led-svetlo-budoucnosti--15726>.
2. *RGBZONE DC5-24V Mini 21 Key RF Remote Controller for WS2811 WS2812B LED Dream Color Strip Light* [online]. Amazon, Inc. [cit. 2022-04-19]. Dostupné z: [https://www.amazon.com/RGBZONE-DC5-24V-Remote-Controller-WS2812B/dp/B075SXMD9Z/ref=sr\\_1\\_22?keywords=RGBZONE&qid=1647956998&sr=8-22](https://www.amazon.com/RGBZONE-DC5-24V-Remote-Controller-WS2812B/dp/B075SXMD9Z/ref=sr_1_22?keywords=RGBZONE&qid=1647956998&sr=8-22).
3. *RGBZONE WS2812B WS2811 WS2801 LED SP108E WiFi Controller DC5 24V for SK6812 SK6812-RGBW WS2812 WS2813 LED Strip Light ,iOS/Android App Group Control* [online]. Amazon, Inc. [cit. 2022-04-19]. Dostupné z: [https://www.amazon.com/RGBZONE-DC5-24V-Controller-SK6812-RGBW-Android/dp/B075SXXXFY/ref=sr\\_1\\_11?keywords=RGBZONE&qid=1647956998&sr=8-11](https://www.amazon.com/RGBZONE-DC5-24V-Controller-SK6812-RGBW-Android/dp/B075SXXXFY/ref=sr_1_11?keywords=RGBZONE&qid=1647956998&sr=8-11).
4. *RGBZONE SP107E Bluetooth LED Music Controller for LED Strip Light WS2811 WS2812B WS2801 SK6812, iOS Android Smartphone App Control* [online]. Amazon, Inc. [cit. 2022-04-19]. Dostupné z: [https://www.amazon.com/dp/B06Y4B171B/ref=emc\\_b\\_5\\_t](https://www.amazon.com/dp/B06Y4B171B/ref=emc_b_5_t).
5. *BTF-LIGHTING K-1000C(T-1000S Updated) WS2812B WS2815 SK6812 WS2811 WS2801 LED Strip LED Matrix LED String 1536 Pixels Controller DC5-24V Addressable Programmable Controller with SD Card* [online]. Amazon, Inc. [cit. 2022-04-19]. Dostupné z: [https://www.amazon.com/BTF-LIGHTING-K-1000C-Controller-Addressable-Programmable/dp/B06XJ1J915?ref\\_=ast\\_sto\\_dp](https://www.amazon.com/BTF-LIGHTING-K-1000C-Controller-Addressable-Programmable/dp/B06XJ1J915?ref_=ast_sto_dp).
6. BLAHOVEC, Antonín. *Elektrotechnika I*. 5., nezměn. vyd. Praha: Informatorium, 2005. ISBN 80-7333-043-1.
7. BLAHOVEC, Antonín. *Elektrotechnika III: (příklady a úlohy)*. 5., nezměn. vyd. Praha: Informatorium, 2005. ISBN 80-7333-045-8.
8. ŠESTÁKOVÁ, Eliška. *Automata and grammars: a collection of exercises and solutions*. 1. vydání. Praha: České vysoké učení technické v Praze, 2018. ISBN 978-80-01-06462-7.
9. *Moore machine* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-04-20]. Dostupné z: [https://en.wikipedia.org/wiki/Moore\\_machine](https://en.wikipedia.org/wiki/Moore_machine).
10. *Secure Shell* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-05-02]. Dostupné z: [https://cs.wikipedia.org/wiki/Secure\\_Shell](https://cs.wikipedia.org/wiki/Secure_Shell).
11. *Pulzně šířková modulace* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-05-02]. Dostupné z: [https://cs.wikipedia.org/wiki/Pulzn%C4%9B\\_%5C%C5%A1%C3%AD%C5%99kov%C3%A1\\_modulace](https://cs.wikipedia.org/wiki/Pulzn%C4%9B_%5C%C5%A1%C3%AD%C5%99kov%C3%A1_modulace).

12. ČSN 33 2000-5-52 ED.2 (332000) *Elektrické instalace nízkého napětí*. 02/2012. Praha: ÚNMZ, 2012.
13. *Raspberry Pi Zero W* [online]. RPishop.cz [cit. 2022-04-19]. Dostupné z: <https://rpishop.cz/zero/647-raspberry-pi-zero-w-4053199547425.html>.
14. *Raspberry Pi Zero 2 W* [online]. Raspberry Pi [cit. 2022-04-19]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>.
15. *Raspberry Pi Zero 2 W Power Consumption* [online]. New York: Future Plc [cit. 2022-05-09]. Dostupné z: <https://cdn.mos.cms.futurecdn.net/UozdccamaqhjkehxyxA2Cd.png>.
16. *WS2812B Datasheet and Specifications* [online]. V1.0. vyd. Dongguan: Worldsemi, 2016 [cit. 2022-05-08]. Dostupné z: [https://voltage.ru/datasheets/WS2812B\\_datasheet\\_EN.pdf](https://voltage.ru/datasheets/WS2812B_datasheet_EN.pdf).
17. *WS2815 datasheet* [online]. V1.1. vyd. Shenzhen: Normand, 2017 [cit. 2022-05-08]. Dostupné z: <http://www.normandle.com/upload/201808/WS2815%5C%20LED%5C%20Datasheet.pdf>.
18. *APA102-202 Super LED* [online]. V1.0. vyd. Shenzhen: LED Color Opto Electronic, 2018 [cit. 2022-05-08]. Dostupné z: <http://www.led-color.com/upload/201604/APA102-2020%5C%20SMD%5C%20LED.pdf>.
19. *WS2815 LED strip* [online]. Dongguang: Dongguan SDIP Light [cit. 2022-05-08]. Dostupné z: <https://www.sdiplight.com/ws2815-led-strip/>.
20. *WS2812 LED strip* [online]. Dongguang: Dongguan SDIP Light [cit. 2022-05-08]. Dostupné z: <https://www.sdiplight.com/ws2812b-led-strip/>.
21. *Difference Between C and Python. Blog* [online]. 2022 [cit. 2022-05-08]. Dostupné z: <https://www.interviewbit.com/blog/difference-between-c-and-python/>.
22. *Apa102-pi* [online]. Switzerland: Tinue [cit. 2022-05-09]. Dostupné z: <https://github.com/tinue/apa102-pi>.
23. *Rpi-ws281x-python* [online] [cit. 2022-05-09]. Dostupné z: <https://github.com/rpi-ws281x/rpi-ws281x-python>.
24. *Zdroj 12V 20,8A 250W spínaný SANPU EPS250-H1V12* [online]. Havlíčkův Brod: ECLIP-SERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/120984-zdroj-12v-20-8a-250w-spinany-sanpu-ps250-h1v12.html>.
25. *Zdroj 5V 7,2A 36W spínaný SANPU PS36-W1V5* [online]. Havlíčkův Brod: ECLIP-SERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/74702-zdroj-5v-7-2a-36w-spinany-sanpu-ps36-w1v5.html>.
26. *Modul světelného senzoru - 3 pin* [online]. České Budějovice: Michal Prenner [cit. 2022-05-09]. Dostupné z: <https://rpishop.cz/senzory/2862-modul-svetelneho-senzoru-3-pin.html>.
27. *Zvukové čidlo* [online]. Havlíčkův Brod: ECLIP-SERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/883-zvukove-cidlo.html>.
28. *IR přijímač s tvarovačem TSOP31236* [online]. České Budějovice: Michal Prenner [cit. 2022-05-09]. Dostupné z: <https://rpishop.cz/senzory/522-ir-prijimac-s-tvarovacem-tsop31236.html>.
29. *JST SM 4P 2.54mm vodiče samec + samice 200mm* [online]. Havlíčkův Brod: ECLIP-SERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/3098-konektor-jst-2.54-sm-4pin-1-par-samec-samice.html>.
30. *Konektor 16mm GX16 - 2 piny - 1 pár* [online]. Havlíčkův Brod: ECLIP-SERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/1986-konektor-16mm-gx16-2-piny-1-par.html>.

31. *Relé 4 kanály 5V s optoclenem* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/2190-rele-4-kanaly-5v-s-optoclenem.html>.
32. *I2C IIC Konvertor Osmikanalový Obousměrný Modul Logic Level* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/1575-i2c-iic-konvertor-osmikanalovy-obousmerny-modul-logic-level.html>.
33. *EMSL.cz: Energetická montážní společnost Liberec, s.r.o.* [Online]. Liberec: EMSL [cit. 2022-05-09]. Dostupné z: [www.emsl.cz](http://www.emsl.cz).
34. *Dvojlinka CYH 2X0,75 ČERNORUDÝ* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/2992-dvojlinka-cyh-2x0-75-cernorudy-vm00vh-h.html>.
35. *Kolébkový vypínač AC 3A/250V* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/7699-kolebkovy-vypinac-ac-3a-250v.html>.
36. *Páčkový přepínač 6A 125VAC* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/1080-packovy-prepinac-6a-125vac.html>.
37. Debugging C/C++ Code on a Raspberry Pi with Visual Studio. *Blog* [online]. 2017 [cit. 2022-05-09]. Dostupné z: <https://dogfeatherdesign.com/2017/01/17/debugging-cc-code-on-a-raspberry-pi-with-visual-studio/>.
38. *SP100/NSP1P PŘÍPOJKOVÁ SKŘÍŇ BEZ PILÍŘE* [online]. ELEKTRO S.M.S., České Budějovice [cit. 2022-05-09]. Dostupné z: <https://shop.elektrosms.cz/cs/sp100-nsp1p-pripojkova-skrin-dck-holoubkov-sk1000128981>.
39. *Trubičkové pojistky rychlé (F) 5x20mm 0.1A-30A sada 150ks* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/75179-trubicrove-pojistky-rychle-f-5x20mm-0.1a-30a-sada-150ks.html>.
40. *Držák pojistek 5x20mm* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/51350-drzak-pojistek-5x20mm.html>.
41. *2.7inch e-Paper HAT* [online]. Shenzhen: Waveshare Electronics [cit. 2022-05-09]. Dostupné z: [https://www.waveshare.com/wiki/2.7inch\\_e-Paper\\_HAT](https://www.waveshare.com/wiki/2.7inch_e-Paper_HAT).
42. *GPIO Rozšiřující Deska pro Raspberry PI 2 3 B B+* [online]. Havlíčkův Brod: ECLIPSERA [cit. 2022-05-09]. Dostupné z: <https://dratek.cz/arduino/1459-gpio-rozsirujici-deska-pro-raspberry-pi-2-3-b-b.html>.



# Obsah přiloženého média

README.txt.....	stručný popis obsahu média
─ bakalarka.rar.....	archiv zdrojového kódu
─ bakalarka.pdf.....	text práce ve formátu PDF