



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**VLÁČEK ŘÍZENÝ MIKROPOČÍTAČEM**

MODEL TRAIN CONTROLLED BY MICROCOMPUTER

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**IVA KAVÁNKOVÁ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Prof. Dr. Ing. PAVEL ZEMČÍK**

BRNO 2019

## Zadání bakalářské práce



21434

Studentka: **Kavánková Iva**  
Program: Informační technologie  
Název: **Vláček řízený mikropočítačem**  
**Model Train Controlled by Microcomputer**  
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte způsob řízení modelu vláčku nebo obdobného modelu mikropočítačem, tedy "embedded počítačem", a to jak literaturu, tak i existující realizace.
2. Navrhněte jednoduchý způsob řízení vláčku a také způsob jeho ovládání uživatelem, zvažte použití například mobilního zařízení jako prostředku uživatelského rozhraní.
3. Popište a diskutujte možnosti a dosažitelné vlastnosti navrženého řešení.
4. Navržený systém implementujte a demonstруйте například na jednoduchém malém kolejišti.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 1. listopadu 2018

## Abstrakt

Tato práce se zabývá řízením modelového kolejiště pomocí kamery. Věnuje se analýze, návrhu a realizaci vestavěného systému. Jedná se o zpracování obrazu v reálném čase a jeho následného vyhodnocení. Navržený a realizovaný produkt tvoří kolejiště se senzorem, k jejichž řízení je použita open-source platforma Arduino Mega 2560 a mikropočítač Raspberry Pi 3 model B. Výsledný systém byl implementován v programovacím jazyku Wiring a ve skriptovacím jazyce Python.

## Abstract

This bachelor thesis is about controlling model trains by camera. It deals with the analysis, design and implementation of the embedded system for modeling track management. The main part is image processing in real-time and following evaluation. Designed and realized product contains model railroad with sensor, which are controlled by open-source platform called Arduino Mega 2560 and microcomputer Raspberry Pi 3 model B. Final system was implemented in programming language Wiring and scripting language Python.

## Klíčová slova

Vlárky, mikropočítač, Arduino, Raspberry Pi, kamera, PWM, modelářství, zpracování obrazu, IoT, uživatelské rozhraní

## Keywords

Trains, microcomputer, Arduino, Raspberry Pi, camera, PWM, modeling, image processing, IoT, user interface

## Citace

KAVÁNKOVÁ, Iva. *Vlárky řízený mikropočítačem*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

# Vláček řízený mikropočítačem

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana profesora Pavla Zemčíka. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....  
Iva Kavánková  
13. května 2019

## Poděkování

Děkuji svému vedoucímu bakalářské práce za cenné rady a odborné vedení.

# Obsah

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Úvod</b>                           | <b>2</b>  |
| <b>2</b> | <b>Řízení modelové železnice</b>      | <b>4</b>  |
| 2.1      | Obecné modelářství . . . . .          | 4         |
| 2.2      | Ovládání kolejiště . . . . .          | 5         |
| 2.3      | Detaily modelových železnic . . . . . | 8         |
| 2.4      | Programy pro návrh kolejišť . . . . . | 10        |
| 2.5      | Platforma Arduino . . . . .           | 11        |
| 2.6      | Arduino Mega 2560 . . . . .           | 15        |
| 2.7      | Platforma Raspberry Pi . . . . .      | 16        |
| 2.8      | Raspberry Pi 3 Model B . . . . .      | 17        |
| 2.9      | Další typy mikrokontrolérů . . . . .  | 18        |
| <b>3</b> | <b>Zhodnocení současného stavu</b>    | <b>20</b> |
| 3.1      | Analýza problému . . . . .            | 20        |
| 3.2      | Vybraný hardware . . . . .            | 21        |
| <b>4</b> | <b>Návrh a implementace</b>           | <b>23</b> |
| 4.1      | Návrh řešení . . . . .                | 23        |
| 4.2      | Návrh modelového kolejiště . . . . .  | 24        |
| 4.3      | Samotné sestavení kolejiště . . . . . | 24        |
| 4.4      | Řídící deska . . . . .                | 26        |
| 4.5      | Snímání kamerou . . . . .             | 29        |
| 4.6      | Zpracování obrazu . . . . .           | 29        |
| 4.7      | Komunikační protokol . . . . .        | 33        |
| 4.8      | Uživatelské rozhraní . . . . .        | 35        |
| 4.9      | Testování . . . . .                   | 36        |
| <b>5</b> | <b>Závěr</b>                          | <b>38</b> |
|          | <b>Literatura</b>                     | <b>39</b> |
| <b>A</b> | <b>Obsah přiloženého CD</b>           | <b>42</b> |
| <b>B</b> | <b>Implementace software</b>          | <b>43</b> |

# Kapitola 1

## Úvod

Vlaky dnes patří mezi nejpopulárnější, nejbezpečnější a nejrychlejší pozemní dopravní prostředky. Mnoho z nás je využívá na každodenní cestování do školy či práce. Není proto divu, že lidé jimi byli fascinováni a rozhodli se vytvářet jejich zmenšené modely a zabývat se modely vláček i ve svém volném čase. Díky tomu vzniklo železniční modelářství, které se díky své rozmanitosti stalo koníčkem napříč generacemi. V modelářství lze realizovat funkční nebo realistické modely, které bývají propracovány do nejmenších detailů. V současnosti existují také soutěže železničních modelářů, které se těší velmi velké oblibě. Velkou nevýhodou tohoto koníčku je velká prostorová (často se jedná i o několik metrů velká kolejiště), časová a finanční náročnost.

Díky moderním technologiím se toto odvětví velmi dynamicky vyvíjí a lze očekávat další inovace a automatizaci. Od počátečního jednoduchého analogového řízení se postupně přechází na digitální řízení. To je však mnohem více finančně náročnější, je nutná DCC (z anglického Digital Command Control) centrála a dekodéry pro lokomotivy a akční prvky. Proto tento typ řízení nebyl použit. V práci byly použity téměř 50 let staré modely a kolejiště. Dokazuje to, že takto lze ovládat naprosto jakékoliv modely vláček, pochopitelně po důkladné revizi – vyčištění a rozebrání některých komponent.

Cílem této bakalářské práce je řízení vláček pomocí mikropočítače a sestavení funkčního kolejiště, které je sledováno kamerou. Ta funguje jako senzor pohybu - monitoruje polohu vláček, dochází k analýze a následnému vyhodnocení obrazu a případnému zabránění kolizí. Jedná se o kombinaci návrhu celého hardwaru a softwaru pro jeho ovládání. Pro větší pohodlí uživatelů bylo vytvořeno jednoduché uživatelské rozhraní, díky kterému lze kolejiště ovládat přes Wi-Fi (z anglického Wireless Fidelity) pomocí například mobilního zařízení.

Ve světě modelářství se pohybuji již od svých dětských let. Mezi mé oblasti zájmu patří již zmíněné vláčky, RC (z anglického Radio Controlled) modely aut a letadel. Pravidelně závodím s RC modely aut. Toto téma je mi velmi blízké a ráda se k němu vracím.

Práce klade důraz na seznámení se s novými typy mikropočítačů, byly vybrány platformy Arduino a Raspberry Pi. Arduino má na starosti kompletní řízení kolejiště (přepínání výhybek a řízení napájení úseků). Bylo vybráno pro své jednoduché použití (nemá operační systém – je to pouze mikrokontrolér) a velký počet pinů, které jsou v případě takto rozsáhlého projektu potřebné. Existuje pro něj spousta dostupných knihoven a podrobná dokumentace. Raspberry Pi bylo vybráno kvůli svému většímu výpočetnímu výkonu, který byl nutný pro snímání a následné zpracování obrazu v reálném čase. Dále také pro větší buffer, aby nebylo nutné načítat fotografie na SD (z anglického Secure Digital) kartu, což by znamenalo další zbytečnou režii.

Tyto platformy musí mezi sebou navzájem komunikovat, proto byl pro jejich komunikaci vytvořen speciální protokol, kterým komunikují přes USB (z anglického Universal Serial Bus) rozhraní. O tomto protokolu blíže pojednává samostatná podkapitola 4.7.

Tato bakalářská práce je rozdělená na několik částí. V následující kapitole 2 jsou zmíněny základní informace o modelářských kolejištích, mikropočítačích, mikrokontrolérech a o již existujících řešeních. Ve 3. kapitole je popsáno zhodnocení současného stavu, specifikace a konečný návrh vlastního řešení. Kapitola 4 podrobně pojednává o popisu vlastní práce od počátečních myšlenek, návrhu, až po samotné sestavení funkčního kolejiště a softwarové zpracování (řízení kolejiště a zpracování obrazu). Pro kolejiště došlo k sestavení hardwaru včetně H-můstků pro napájení úseků a relé pro přepínání výhybek. Při snímání obrazu dochází ke zpracování jednotlivých snímků v nekonečné smyčce. V případě blížící se srážky dojde k zastavení vlaků. V závěru (v kapitole 5) je zhodnocena celá práce a také uvedeny možnosti, jak lze na tuto práci navázat.

## Kapitola 2

# Řízení modelové železnice

Tato kapitola pojednává o shrnutí současného stavu. Zabývá se modelářskými kolejišti, ovládáním a počítačovými programy pro jejich návrh. Také popisuje platformy Arduino a Raspberry Pi, které porovnává s jinými typy mikrokontrolérů.

### 2.1 Obecné modelářství

Pojmem modelářství se označuje výroba modelů, existujících nebo i plánovaných objektů, staveb, strojů a další. Velmi důležitou roli v modelářství hraje měřítko, neboli poměr mezi velikostí předlohy a daného modelu. Modelů se využívá i v architektuře, vědeckých výzkumech ale i v dalších oblastech lidské činnosti. [9]

Modely dělíme na:

- Rádiem řízené modely (RC modely),
- modelové železnice,
- autodráhy,
- papírové modely (vystřihovánky),
- plastické modely,
- a další...



Obrázek 2.1: Největší modelové kolejiště v ČR - Království železnic v Praze.<sup>1</sup>

<sup>1</sup>Převzato z: <https://www.zelpage.cz/story/modely/p7010074.jpg>



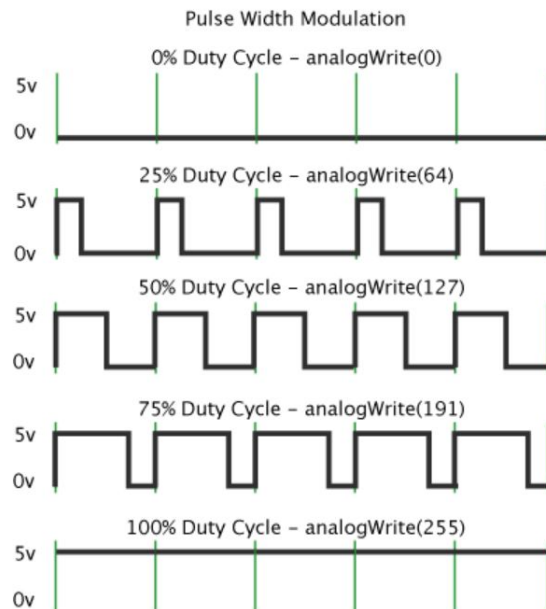
## 2.2 Ovládání kolejiště

### Pulzně šířková modulace - PWM

Pulzně šířková modulace (PWM – z anglického Pulse Width Modulation) je diskretní modulace pro přenos analogového signálu pomocí digitálního signálu. Vzhledem ke svým specifickým vlastnostem se často využívá v elektronice pro řízení velikosti potřebného proudu nebo napětí. [13]

Samotný signál, může nabývat hodnot logické 0 a 1 (tzn. vypnuto a zapnuto). Hodnota přenášeného signálu je zakódována jako poměr mezi stavy zapnuto/vypnuto - nazývá se střída. V zahraniční literatuře bývá tento pojem označován jako Duty cycle [29]. Délka pulzu ve stavu logické 1 se nazývá šířkou pulzu. K získání potřebných analogových hodnot napětí nebo proudu je tato šířka měněna.

Mezi nejjednodušší příklad použití patří nastavení jasu LED (z anglického Light Emitting Diode) diody. Logická 0 značí zhasnutou LED diodu, logická 1 značí rozsvícenou LED diodu. Změnou střídy se mění intenzita jasu.



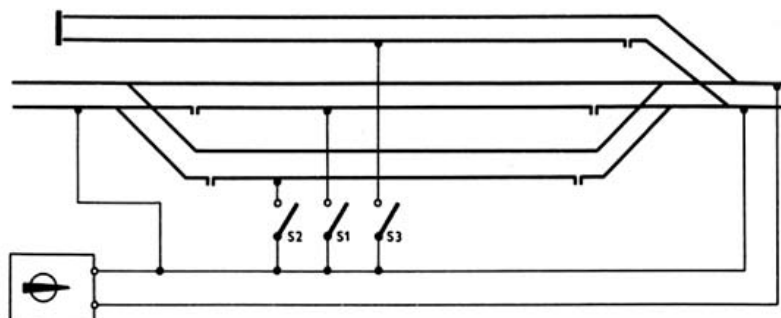
Obrázek 2.2: Ukázka průběhu pulzně šířkové modulace.<sup>2</sup>

Obrázek 2.2 popisuje signál s různými střídami. Jedná se o použití pulzně šířkové modulace u platformy Arduino, která pro tyto účely používá funkci `analogWrite()`. Parametrem této funkce jsou hodnoty v rozmezí 0 – 255. To znamená, že `analogWrite(0)` je signál s 0% střídou. Naopak `analogWrite(255)` je signál se 100% střídou - odpovídá tomu i daný časový průběh na obrázku.

<sup>2</sup>Převzato z [22]

## Analogové ovládání

Ovladač vlaku (transformátor s regulátorem) přivádí do kolejí stejnosměrné napětí v rozmezí 0 – 12 V a rychlost vlaku je ovlivňována velikostí napětí. Při takovém řízení lze na jednom kolejišti jezdit pouze s jednou lokomotivou. V případě, že požadujeme jízdu více vlaků najednou, je nutné rozdělit kolejiště na jednotlivé úseky, které nejsou vodivě spojeny a mají své vlastní napájení.



Obrázek 2.3: Příklad jednoduchého zapojení pro analogové ovládání.<sup>3</sup>

## Digitální ovládání - DCC

S rozvojem elektrotechniky došlo k inovaci řízení, kdy je každá lokomotiva vybavena dekodérem, který vyhodnocuje, zda-li jsou příkazy z ovladače určeny právě pro konkrétní model. V kolejích je trvalé napětí a zároveň kolej slouží jako sběrnice, kterou se přenáší dané příkazy z centrálního ovládání do jednotlivých lokomotiv.

Současně lze jezdit s více vlaky a řídit také jejich další funkce: pouštět zvuky, rozsvěcovat světla. Mezi velkou nevýhodou patří velmi vysoká pořizovací cena. [10]

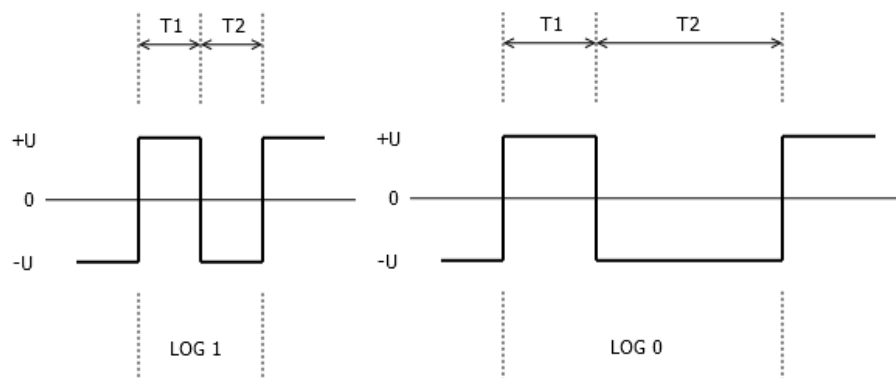
## Popis ovládání DCC

Standard DCC (z anglického Digital Command Control) je celosvětově používaný pro řízení modelových kolejišť. Byl vyvinut v 80. letech minulého století (německou firmou Lenz Elektronik GmbH) pro dva německé výrobce železnic Märklin and Arnold. Spravuje jej americká organizace NMRA (z anglického National Model Railroad Association).

V tomto systému se využívá dvouvodičové komunikace střídání polarity napájecího napětí. V kolejích je nepřetržité napětí a zároveň se po kolejích posílá signál, který ovládá jednotlivé lokomotivy s dekodéry (ty reagují podle své adresy). Tyto digitální dekodéry je nutno umístit ke každému akčnímu prvku (lokomotiva, výhybka,...). Díky tomuto systému je možno snadno měnit rychlost jízdy vlaku a jeho směr.

<sup>3</sup>Převzato z [4]

Samotný digitální signál je kódován pomocí šířky impulsu. Při posílání logické jedničky musí být časy  $T_1$  a  $T_2$  shodné (v rozmezí 52 - 64  $\mu\text{s}$ ), v případě posílání logické nuly mohou být časy  $T_1$  a  $T_2$  různé (v rozmezí 95 - 9900  $\mu\text{s}$ ). [11]



Obrázek 2.4: Posílání logické jedničky a logické nuly.<sup>4</sup>

Tento signál je současně na obou kolejích, avšak s opačnou polaritou signálu.

## DCC pakety

Komunikace probíhá skrze DCC pakety, které se skládají z několika kódovaných bloků. Stavba paketu:

- **Preamble**  
Posloupnost minimálně deseti po sobě jdoucích jedniček, slouží pro určení začátku celého paketu.
- **Start Bit**  
Vložena logická nula, pro k oddělení jednotlivých částí paketu.
- **Adresa**  
Určuje adresu lokomotivy nebo jiné periferie (první byte).
- **Data**  
Obsahuje data pro konkrétní lokomotivu nebo periférii (druhý byte).
- **Kontrola**  
Počítá se jako funkce exkluzivní OR adresového bytu s datovým bytem (třetí byte).
- **Stop Bit**  
Vložena logická jednička, udává konec paketu.

<sup>4</sup>Převzato z [11]

## Paket pro lokomotivu

Jedná se o paket pro rychlost a směr jízdy, který využívají lokomotivy.

| Preamble     | Start Bit | Adresa   | Start Bit | Data     | Start Bit | Kontrola | Stop Bit |
|--------------|-----------|----------|-----------|----------|-----------|----------|----------|
| 111111111111 | 0         | 0AAAAAAA | 0         | 01DUSSSS | 0         | EEEEEEEE | 1        |

Tabulka 2.1: Příklad paketu pro rychlost a směr jízdy.<sup>5</sup>

Vysvětlivky:

**0** - logická nula

**1** - logická jednička

**A** - určuje adresu lokomotivy 1 – 127

**S** - nastavuje rychlost

**D** - udává směr

**E** - kontrolní bity, vzniknou pomocí funkce exkluzivní OR (adresy a dat)

Existuje mnoho typů paketů, všechny jsou popsány ve standardu NMRA. [5]

## DCC centrála

Pro digitální řízení je nutná DCC stanice, která má za úkol modulovat DCC signál a odesílat jej do zesilovačů, které poté napájí koleje. Tato stanice je založena na mikrokontroléru či mikropočítači, který obsluhuje ostatní komponenty – výkonný zesilovač DCC pro napájení kolejí, systémové sběrnice pro ovládání periferií.

## DCC++

DCC++ je open-source projekt založen na platformě Arduino. Umožňuje vytvořit si vlastní DCC ovládací stanici. Jádrem je mikrokontrolér Arduino UNO nebo MEGA, dále je nutný Arduino Motor Shield R3. Ovládání centrály je možné přes USB port, bezdrátově pomocí Bluetooth nebo po síti (nutný Ethernet Shield). Další částí projektu je DCC++ kontrolér, což je GUI (z anglického Graphical User Interface), které slouží pro rozvržení kolejiště a jeho následné ovládání. [12]

## 2.3 Detaily modelových železnic

### Měřítko

Během vývoje vlakového modelářství došlo ke vzniku mnoha měřítek modelových železnic. Měřítko se udává v poměru s reálným modelem. S každým měřítkem je spjat i rozchod kolejí.

---

<sup>5</sup>Převzato z [8]

Nejpoužívanější měřítko přehledně popisuje následující tabulka:

| Velikost | Měřítko  | Rozchod kolejí |
|----------|----------|----------------|
| II       | 1 : 22,5 | 63,5 mm        |
| I        | 1 : 32   | 45 mm          |
| 0        | 1 : 45   | 32 mm          |
| S        | 1 : 64   | 22,5 mm        |
| H0       | 1 : 87   | 16,5 mm        |
| TT       | 1 : 120  | 12 mm          |
| N        | 1 : 160  | 9 mm           |
| Z        | 1 : 220  | 6,5 mm         |

Tabulka 2.2: Tabulka měřítek železničních modelů.<sup>6</sup>

V současnosti mezi nejoblíbenější měřítka patří:

- **Velikost TT** – zkratka TT (z anglického Table Top) - v překladu znamená deska stolu, nám názorně říká, že si lze postavit kolejíště desce běžného stolu [7]. Mezi výhody této velikosti patří především zřejmý menší požadavek na prostor.
- **Velikost H0** – v současnosti je tato modelová velikost nejvíce rozšířena a vyrábí ji většina výrobců. K jejímu vybudování však potřebujeme velmi mnoho místa v řádu několika metrů. Zde lze také pozorovat větší robustnost modelů (než u TT), což ocení především děti.

## Typy kolejí

- **Plechové koleje** – jsou nejjednodušší a také nejlevnější. Jsou vyrobeny z ohnutého ocelového plechu. Vedení vlaku není příliš přesné (vlak může sám vykolejit) a v případě oxidace je již nelze čistit.
- **Profilové koleje** – jsou vyrobeny z plného materiálu. Jejich výhodou oproti plechovým kolejím je to, že zajišťují spolehlivější vedení vlaku a také je lze v případě oxidace úspěšně vyčistit.
- **Profilové koleje s podložím** – jedná se o profilové koleje doplněné o většinou plastové podloží, které je zpevní a zároveň zajišťuje spolehlivé elektrické propojení jednotlivých kolejnic a to i v případě opakovaného rozpojování.

## Modely vláček

Většina modelů je zmenšeninou skutečné lokomotivy nebo vagónu včetně sebemenších detailů. Díky tomu jsou modely vláček velmi blízko realitě. Mezi modely lze také najít sběratelské kusy, od prvorepublikových vozů až po moderní japonské vlaky Šinkansen [6]. Ceny jedinečných kusů se mohou dosahovat až několik stovek tisíců.

---

<sup>6</sup>Převzato z [31]

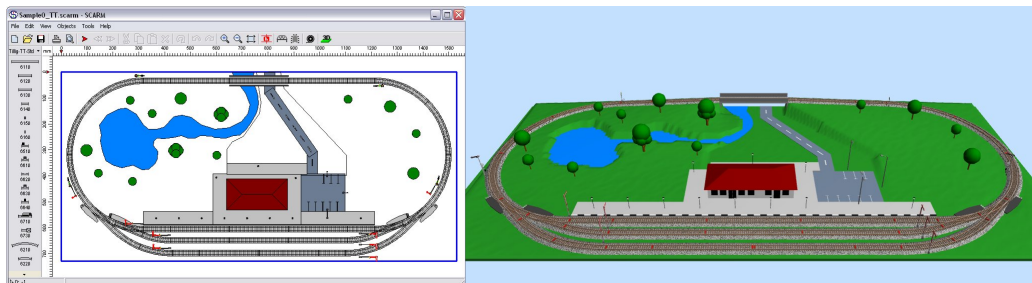


Obrázek 2.5: Sběratelský kus - elektrická lokomotiva E499.119 ČSD (H0).<sup>7</sup>

## 2.4 Programy pro návrh kolejišť

### SCRAM – Simple Computer Aided Railway Modeller

SCRAM je bezplatný program, který umožňuje jednoduše vytvářet a modelovat složitější kolejiště. Umožňuje komplexní návrh včetně následné vizualizace vytvořeného kolejiště. Součástí tohoto softwaru je více než 195 knihoven obsahující nejrůznější objekty a modely od světových firem. [25]



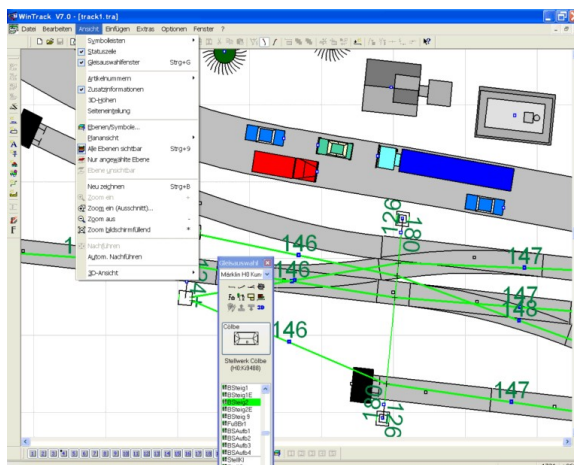
Obrázek 2.6: Návrh a následná vizualizace v prostředí SCRAM.<sup>8</sup>

### WinTrack

WinTrack je komerční program, používaný pro modelování kolejišť. Umožňuje návrhy až do rozměrů kolejiště 50 x 50 metrů. [19] Kromě obsáhlých knihoven také nabízí možnost přidání vlastních prvků – typů kolejí, příslušenství, vozidel a terénu. Dále umí také hlídat kritické vzdálenosti a výšky. V neposlední řadě také dokáže vygenerovat podrobný kusovník i s cenou použitých prvků.

<sup>7</sup>Převzato z [3]

<sup>8</sup>Převzato z: <https://www.scarm.info/index.php?page=screenshots>

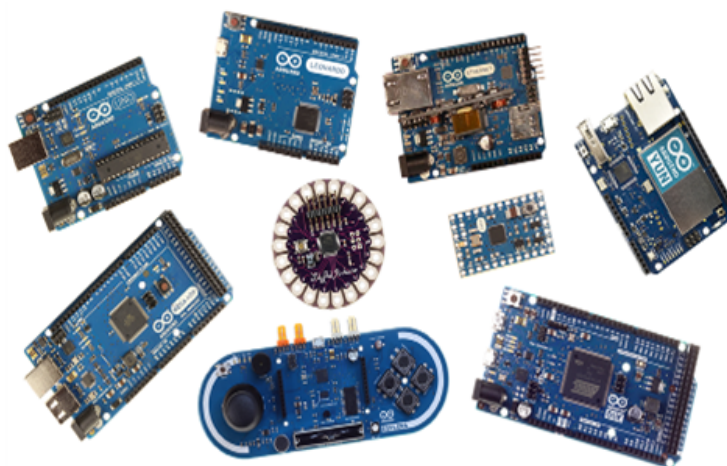


Obrázek 2.7: Ukázka práce v softwaru WinTrack.<sup>9</sup>

Tento program pracuje ve vrstvách, tudíž neumí jen návrh samotného vizuálního vzhledu, ale je možno vytvořit i vrstvy s kabeláží. Po zakreslení návrhu dokáže program spočítat a zobrazit vygenerovaný 3D model kolejíště, čímž nám pomáhá s představivostí, která může být (především) u velkých kolejíšť problém [28].

## 2.5 Platforma Arduino

Arduino je open-source platforma s grafickým vývojovým prostředím, která klade důraz na jednoduchost použití. Tento projekt vznikl v roce 2005 v městě Ivrea v Itálii. Jeho hlavním cílem bylo vytvořit jednoduchou platformu pro studenty, umožňující rychlý vývoj a jednoduché používání. Jeho zakladatelé David Cuartielles a Massimo Banzi svůj projekt pojmenovali po Arduinovi Ivrejském, významné historické postavě města [14].



Obrázek 2.8: Různé typy Arduino desek.<sup>10</sup>

<sup>9</sup>Převzato z: <https://i.iinfo.cz/s/4/64665-wintrack.jpg>

<sup>10</sup>Převzato z: <https://www.elprocus.com/different-types-of-arduino-boards/>

Pomocí mikropočítače Arduino lze číst vstupy (pomocí funkcí `digitalRead()` – pro čtení z digitálních vstupů a `analogRead()` – pro čtení analogových vstupů), jako například stisknutí tlačítka, nadefinování výstupů, aktivaci motoru nebo rozsvícení LED diody. Program lze na desku nahrát jednoduše z počítače pomocí USB kabelu. K programování slouží programovací jazyk Wiring, který je založen na programovacím jazyce C++ [26]. Jako vývojové prostředí lze použít originální Arduino IDE (z anglického Integrated Development Environment) nebo se například nabízí možnost stažení pluginu například do vývojového prostředí Microsoft Visual Studio (pro pokročilejší uživatele, zde je zde však možnost využití jednoduchého debuggeru pro ladění programů a hledání chyb) [15].

## Programování Arduina

Arduino je možné programovat v jazyce C++ (framework Wiring). Celý program je postaven na následující specifické struktuře.

Program je rozdělen do dvou specifických bloků programu. V první části je to funkce `void setup()`, která se provede pouze jednou na začátku běhu programu. Nachází se v ní například inicializace pinů, otevření sériového portu a podobně - věci co potřebujeme jednou jednou za dobu běhu programu. Ve druhé části máme funkci `void loop()`, která se provádí v nekonečné smyčce až do odpojení napájení (či zmáčknutí tlačítka reset) - programujeme konkrétní události (například blikání LED diody).

Časové prodlevy (kdy Arduino pozastaví svou činnost) lze zadat funkcí `delay()`, která jako parametr bere čas v milisekundách. Například `delay(42)` znamená, že Arduino po dobu 42 ms nebude vykonávat žádnou činnost.

Zde je příklad jednoduchého programu, který pouze bliká s LED diodou:

```
1  void setup() {
2      pinMode(12, OUTPUT); //nastav pin 12 jako vystup
3  }
4
5  void loop() {
6      digitalWrite(12, HIGH); //na pinu 12 pust proud
7      delay(1000); //pockej 1000 ms = 1 s
8      digitalWrite(12, LOW); //na pinu 12 vypni proud
9      delay(1000);
10 }
```

Obrázek 2.9: Jednoduchý program pro blikání s LED diodou.<sup>11</sup>

---

<sup>11</sup>Převzato z [38]



## Typy desek Arduino

Firma Arduino nabízí mnoho typů vývojových desek, které se liší velikostí, čipem, počty pinů a podobně. V současnosti existuje v oficiální nabídce více než 20 typů desek [17]. Jsou rozděleny do kategorií podle možnosti využití - základní modely pro začátečníky, doplňkové moduly, moduly pro internet věcí. Lze také provést podrobné porovnání detailů a podle toho vybrat nejvíce vhodnou desku pro náš projekt [18].

- **Arduino UNO**

V současné době je nejvíce používaná deska, je vhodná i pro začínající uživatele. Je osazena běžným procesorem ATmega 328 a má klasické USB. Současná verze desky je Rev3. Vyvinuly se z ní další desky - Arduino Ethernet (vybavené Ethernet portem pro připojení k síti) a Arduino Bluetooth (vybavené technologií Bluetooth pro bezdrátovou komunikaci).

- **Arduino Mega**

Větší deska, díky tomu má prostor pro větší a výkonnější čipy a také pro více pinů. Této desce je věnována samostatná kapitola 2.6.

- **Arduino Due**

Nejvýkonnější deska, jako jediná je založená na 32bitové architektuře s větší pamětí. Na rozdíl od ostatních desek Arduino běží na 3,3 V. Deska má dva microUSB konektory (jeden pro programování, druhý pro připojení dalších zařízení).

- **Arduino Mini**

Nejmenší ze všech oficiálních Arduino desek a byl navržen pro úsporu místa. Tato verze nedisponuje USB portem, tudíž je nutno použít externí převodník. Deska běží na procesoru ATmega328 s taktem 16MHz. Je určena pro chytré vypínače, dálkové ovladače a obecně všechny zařízení, kde je zapotřebí co nejmenší deska.

- **Arduino Nano**

Další z malých desek, narozdíl však od Arduino Mini obsahuje USB port, je o něco málo větší. Poskytuje větší komfort, neboť se nemusí používat externí převodník. Disponuje stejným čipem jako Arduino Mini, takže jeho výkonnost je stejná.

- **Arduino Micro**

Jedna z mála desek, která obsahuje převodník v čipu. Pro počítač, ke kterému je tato deska připojena se tváří jako myš a nebo klávesnice. Proto je vhodná pro uživatele, kteří si chtějí sami vytvořit takové zařízení, které může tak počítači zasílat příkazy jako stisk klávesy nebo posunutí myši bez nutnosti přeprogramování převodníku.

- **Lilypad Arduino**

Speciální typ desky, u níž jsou spoje tvořeny vodivou nití a díky tomuto je tato deska uzpůsobena pro nošení na textilu. Můžeme se setkat s verzí s USB a čipem ATmega32u4, nebo bez USB ve verzi ATmega328 a podobně. Dá se použít například pro vytvoření cyklistické mikiny s přišitými blinkry, pro větší bezpečnost na cestách.

- **Arduino Robot**

Deska vytvořená pro sestavení vlastního robota. Je obsazena čipem ATmega32u4. Zajímavostí je přítomnost kompasu.

- **Arduino Esplora**

Hybridní deska, která obsahuje joystick, tlačítka a posuvný potenciometr. Nalezneme zde také piezo měnič, teploměr, tříosý akcelerometr nebo piny pro připojení LCD (z anglického Liquid Crystal Display) displeje. Díky těmto komponentám se z něj dá vytvořit samostatný herní set nebo konzole pro hraní her. Obsahuje čip ATmega32u4 [36].

## Arduino Shieldy

Arduino Shieldy jsou rozšiřující karty pro desky Arduino. Existuje jich celá řada, například Ethernet Shield, Wifi Shield, Motor Shield a podobně. Důležité je si však dát pozor, aby Shield byl kompatibilní s použitou verzí desky Arduino.

Příklady Arduino Shieldů [1]:

- **Arduino Shield Wi-Fi**

Umožňuje bezdrátovou komunikaci přes Wi-Fi na frekvenci 2,4 GHz ve standardu 802.11b/g.

- **Arduino Shield – Motor modul**

Pro připojení motorů s pracovním napětím 5V – 12V a maximálním proudem pro jeden motor je 2A.

- **Arduino Ethernet Shield**

Přidává možnost komunikace prostřednictvím LAN (z anglického Local Area Network) sítě.

- **Arduino Relay Shield**

Spíná pomocí relé nezávislé obvody.

- **Arduino GSM Shield**

Komunikuje prostřednictvím GSM (z francouzského Groupe Spécial Mobile) sítě.

## Arduino klony

Kromě oficiální řady desek Arduino, existují také neoficiální klony. Arduino klon lze poznat většinou podle názvu, který obsahuje příponu -duino (celé slovo Arduino výrobci použít nemohou, je totiž chráněno licenci) [37]. Mezi jejich výhody patří především levnější cena či jiný typ. Často používají jiný USB převodník a je nutné si nainstalovat potřebný ovladač. Dalším problémem bývá menší životnost. Většina klonů je kompatibilní se softwarem Arduino IDE (z anglického Integrated Development Environment), je pouze nutné v nastavení změnit typ desky a případně postahovat jiné knihovny [34].

Příklady klonů [36]:

- **ArduPilot**

Klon navržený pro ovládání autonomních létajících zařízení (letadla, kvadrokoptéry,...).

- **Rainbowduino**

Určeno pro připojení a následného řízení 8×8 maticového RGB LED displeje.

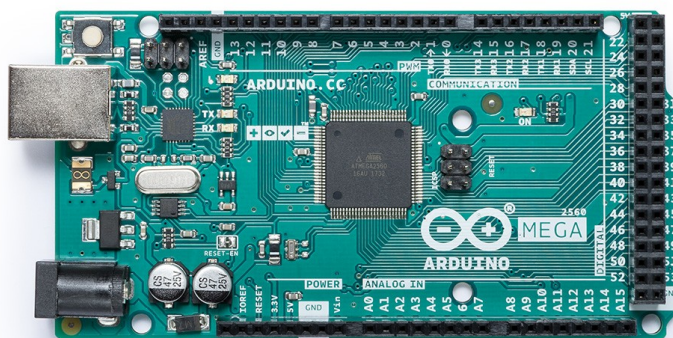
- **Freaduino, Seeeduino**

Levnější kopie originálních desek.

- **Wemos**  
Desky s uplatněním především v IoT (z anglického Internet of Things), využívá ESP čipy [34].

## 2.6 Arduino Mega 2560

Arduino Mega 2560 bylo navrženo pro komplexnější projekty, které potřebují více vstupně/výstupních pinů, více paměti programové paměti a paměti RAM, než poskytují ostatní desky. Díky tomu je vhodná především pro 3D tiskárny a robotiku [16].



Obrázek 2.10: Detail desky Arduino Mega 2560.<sup>12</sup>

Disponuje 54 digitálními I/O piny (12 z nich se umí chovat jako PWM výstupy), 16 analogovými vstupy, 4 hardwarovými sériovými porty UART (z anglického Universal Asynchronous Receiver-Transmitter), krystalovým oscilátorem o frekvenci 16 MHz, USB portem, napájecím konektorem, konektorem ICSP (z anglického In-Circuit Serial Programming) a resetovacím tlačítkem [2].

Předchozí verzí bylo Arduino Mega 1280, které disponovalo procesorem ATmega1280 (odtud odvozen název), díky jeho popularitě a pozdějšímu rozvoji technologií byl nahrazen nynější verzí Arduino Mega 2560 s výkonnějším procesorem ATmega2560, větší pamětí FLASH a také menší spotřebou [33].

<sup>12</sup>Převzato z [16]

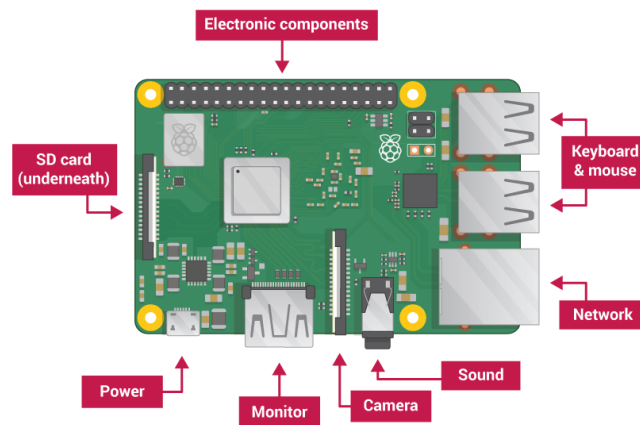
|   |  |
|---|--|
| Mikrokontrolér                                | ATmega2560                             |
| Provozní napětí                               | 5V                                     |
| Doporučené vstupní napětí                     | 7 – 12V                                |
| Rozsah vstupního napětí                       | 6 – 20V                                |
| Digitální vstupně/výstupní piny               | 54 (z toho 12 poskytuje výstup PWM)    |
| Analogové vstupní piny                        | 16                                     |
| Stejnoseměrný proud na vstupně/výstupním pinu | 20 mA                                  |
| Stejnoseměrný proud pinu s napětím 3,3V       | 50 mA                                  |
| Paměť Flash                                   | 256 kB, z toho 8 kB využívá bootloader |
| Paměť SRAM                                    | 8 kB                                   |
| Paměť EEPROM                                  | 4 kB                                   |
| Rychlost hodin                                | 16 MHz                                 |
| Délka   | 101,52 mm                              |
| Šířka   | 53,3 mm                                |
| Hmotnost                                      | 37 g                                   |

Tabulka 2.3: Tabulka technické specifikace desky Arduino mega 2560 Rev3.<sup>13</sup>

## 2.7 Platforma Raspberry Pi

Raspberry Pi je malý jednodeskový počítač o velikosti standardní platební karty. Byl vyvinut v roce 2012 britskou nadací Raspberry Pi Foundation s cílem podpořit výuku informatiky na školách. Má také operační systém, primárním operačním systémem je Raspbian. Existují však i další operační systémy jako Windows IoT Core, Ubuntu MATE [32].

Jedná se o jednočipový počítač, který je však srovnatelný se méně výkonným stolním počítačem. Obsahuje konektor HDMI (z anglického High-Definition Multi-media Interface) pro monitor a USB porty (pro připojení klávesnice a myši). Po připojení k síti je možno se na něj lze dostat přes protokol SSH (z anglického Secure Shell) a pracovat vzdáleně.



Obrázek 2.11: Detailní popis mikropočítače Raspberry Pi.<sup>14</sup>

<sup>13</sup>Převzato z [16]

<sup>14</sup>Převzato z: <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/4>

U Raspberry Pi rozlišujeme tři generace, původní (z roku 2012), druhou generaci Raspberry Pi 2 (z roku 2015) a třetí generaci Raspberry Pi 3 (z roku 2016) o které podrobně pojednává kapitola 2.8 (konkrétně o modelu Raspberry Pi 3 Model B). Typy modelů [23]:

- Raspberry Pi Model A – z roku 2013, 256 MiB RAM (z anglického Random Access Memory) sdílených s grafickou kartou, slot pro SD nebo MMC kartu, jeden USB port.
- Raspberry Pi Model A+ – z roku 2014, 256 MiB RAM sdílených s grafickou kartou, slot pro SD kartu, jeden USB port.
- Raspberry Pi Model B – z roku 2012, 512 MiB RAM sdílených s grafickou kartou, slot pro SD nebo MMC (z anglického MultiMediaCard) kartu, dva USB porty.
- Raspberry Pi Model B+ – z roku 2014, 512 MiB RAM sdílených s grafickou kartou, slot pro SD, čtyři USB porty, ethernetový adaptér 10/100 s konektorem RJ-45.

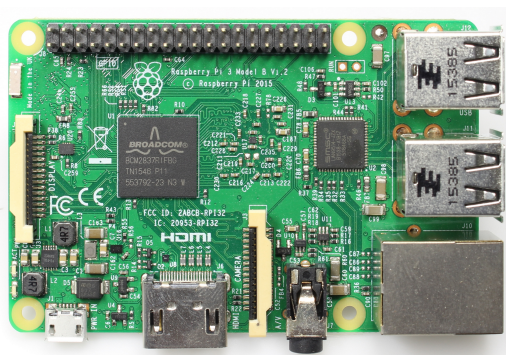
## Rozdíly mezi Arduinem a Raspberry Pi

Hlavní rozdíl mezi těmito platformami je ten, že Arduino je pouze mikrokontrolér a Raspberry Pi je mikropočítač (má tedy i operační systém). Arduino se používá na jednodušší, méně výpočetně náročné úkoly, které provádí prostřednictvím jednoho programu pořád dokola. Raspberry Pi je plnohodnotný počítač s větším výpočetním výkonem, na kterém může běžet více programů. Je také mnohem složitější na použití než Arduino [30]. Nemalý rozdíl je také v pořizovací ceně.

## 2.8 Raspberry Pi 3 Model B

Mikropočítač Raspberry Pi 3 Model B je jeden z nejpokročilejších mikropočítačů na trhu. Mezi jeho přednosti patří [24]:

- Vyšší frekvence procesoru (nárůst o 33 % oproti minulým verzím) – 1,2 GHz,
- 64-bitová architektura,
- instrukční sada ARMv8 – širší podpora GNU/Linux systémů.



Obrázek 2.12: Mikropočítač Raspberry Pi 3 Model B.<sup>15</sup>

<sup>15</sup>Převzato z: [https://upload.wikimedia.org/wikipedia/commons/4/49/Raspberry\\_pi\\_3\\_model\\_b\\_v1.3\\_bot.jpg](https://upload.wikimedia.org/wikipedia/commons/4/49/Raspberry_pi_3_model_b_v1.3_bot.jpg)

Podrobnější specifikace v následující tabulce:

|                                   |                           |
|-----------------------------------|---------------------------|
| Mikrokontrolér                    | BCM2837                   |
| Procesor                          | Quad Cortex A53 @ 1,2 GHz |
| Instrukční sada                   | ARMv8-A                   |
| Grafický procesor                 | 400 MHz VideoCore IV      |
| Paměť RAM                         | 1 GB SDRAM                |
| Úložný prostor                    | micro-SD                  |
| Ethernet                          | 10/100                    |
| Bezdrátové připojení              | 802.11n / Bluetooth 4.0   |
| Video výstup                      | HDMI / kompozitní         |
| Audio výstup                      | HDMI / sluchátka          |
| Univerzální vstupně/výstupní piny | 40                        |

Tabulka 2.4: Technická specifikace mikropočítače Raspberry Pi 3 Model B.<sup>16</sup>

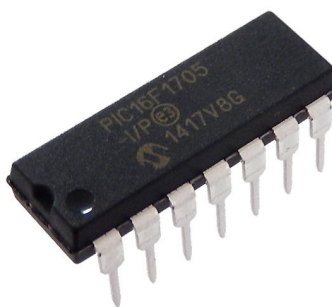
Z těchto technických parametrů vyplývá, že tento mikropočítač je na svou velikost velmi výkonný a tudíž vhodný pro výpočetně náročnější projekty.

## 2.9 Další typy mikrokontrolérů

Tato podkapitola pojednává o dalších vybraných mikropočítačích, které by bylo možné použít pro řízení vláčku. Je doplněna o srovnání s platformami použitými v této bakalářské práci.

### Mikrokontroléry PIC

Mikrokontroléry PIC (z anglického Peripheral Interface Controller) jsou programovatelné jednočipové počítače od firmy Microchip Technology sídlící v USA. Tyto programovatelné součástky jsou určeny pro nejrůznější řídicí úkony v průmyslu, pro řídicí a měřicí systémy a podobně. Využívají Harvardskou architekturu (oddělené paměti programu a dat) a mají redukovanou instrukční sadu – RISC (z anglického Reduced Instruction Set Computer) [21].



Obrázek 2.13: Mikrokontrolér PIC16F1705.<sup>17</sup>

<sup>16</sup>Převzato z [27]

<sup>17</sup>Převzato z [35]

Výhoda platformy Arduino oproti mikrokontrolérům PIC je především v tom, že je dnes ve světě velmi rozšířená a díky tomu je levnější. Díky velké komunitě uživatelů poskytujících různé příklady a návody, obsáhlé dokumentaci a spoustou knihoven je snadnější její používání. Arduino také nabízí kompletní desky a není nutné kupovat žádné jiné nutné součásti, na rozdíl od mikrokontrolérů PIC.

## **FITkit**

FITkit je hardwarová platforma vyvinutá na Fakultě informačních technologií VUT v Brně [20]. Vznikla pro potřeby výuky, aby si studenti vyzkoušeli své teoretické znalosti v praxi. Tvoří jej výkonný procesor MPS430, hradlové pole FPGA (z anglického Field Programmable Gate Array) a řada periférií – klávesnice, LCD display, audio rozhraní a další. Pro účely programování vzniklo vývojové prostředí QDevKit. Celý tento projekt je koncipován jako open-source a open-core [20].



Obrázek 2.14: FITkit verze 2.0.<sup>18</sup>

Tato platforma nebyla využita z důvodu jejího používání ve výuce na univerzitě, jelikož cílem této bakalářské práce je seznámit se a použít další typy mikrokontrolérů/mikropočítačů.

---

<sup>18</sup>Převzato z: <http://merlin.fit.vutbr.cz/FITkit/imgs/uvod/001.png>

## Kapitola 3

# Zhodnocení současného stavu

Cílem práce bylo prostudovat a navrhnout způsob řízení modelů vláčku pomocí mikropočítače a jeho ovládání uživatelem. Následně daný systém implementovat a demonstrovat na modelu kolejiště.

Při výběru mikropočítačů jsou zvoleny dvě platformy – Arduino a Raspberry Pi. Komunikují mezi sebou prostřednictvím vlastního komunikačního protokolu. Došlo také k přidání senzoru pro snímání obrazu, který je nad kolejištěm. Jedná se o kameru, která kolejiště fotí a následně dojde k vyhodnocení situace na kolejišti.

### 3.1 Analýza problému

V první řadě bylo nutné si blíže upřesnit, co bude vlastním výsledkem práce, neboť dané zadání nebylo příliš konkrétní, a bylo nutné blíže specifikovat výsledek práce.

Mohlo se jednat o obyčejné oválné kolejiště bez jakýchkoliv periférií. Na základě přiblížení se k realitě skutečných modelářských kolejištím, bylo rozhodnuto postavit takové kolejiště, na kterých bude možné demonstrovat situace odpovídající skutečnosti – slepou kolej, výhybky a nádraží. Principy této práce budou aplikovatelné i na jakékoliv jiné modelové kolejiště.

I když zadání definovalo pouze mikropočítač, došlo k rozšíření na dvě různé platformy (Raspberry Pi a Arduino), které mezi sebou budou komunikovat. Mohlo se jednat o komunikaci přes síť, což by Raspberry Pi bez problému zvládlo, ale k Arduinu by musel být přidán rozšiřující Ethernet Shield. Proto bylo rozhodnuto o vzájemné komunikaci přes USB rozhraní, kterým oba mikropočítače disponují. Budou si posílat mezi sebou speciální sekvence znaků pro řízení modelového kolejiště.

Arduino bude mít na starosti kompletní ovládání kolejiště, přepínání výhybek a napájení úseků. Raspberry Pi tomu bude řídit kameru a následné zpracování obrazu. Po vyhodnocení situace pošle Arduinu zprávu o tom má dělat. Jako komunikační protokol byl zvolen řetězec znaků, kde prvních šest číslic reprezentuje napájení úseků a další tři číslice označují stav výhybek. Tato čísla budou nabývat více hodnot (jízda po/proti směru hodinových ručiček, přepínání výhybek rovně/do odbočky). V případě potřeby bude možné tento protokol jednoduše rozšířit - přidáním dalších znaků pro další věc (periferii, úsek a podobně).

Bylo dohodnuto, že dojde k přidání senzoru (kamery), která bude sledovat pohyb na kolejišti. Kamera bude umístěna uprostřed nad kolejištěm v takové výšce, aby snímala celé modelové kolejiště. Pro tyto účely byl zakoupen modul s kamerou s vyměnitelnými objektivy.



Pro zpracování obrazu a nalezení kolejiště byly ke kolejím přidány speciální kalibrační značky, které pomohou hledat samotné koleje a dělit kolejiště na podúseky pro hledání následné polohy lokomotiv.

Díky velké finanční náročnosti a stáří lokomotiv bylo rozhodnuto, že nebude použito modernější DCC ovládání, ale starší analogové, které lokomotivy bez újmy zvládnou. Vzhledem k zvolenému typu ovládání bude nejprve nutné modelové kolejiště rozdělit na jednotlivé napájecí úseky.

Vzhledem ke tvaru kolejiště je určeno šest napájecích úseků. Dva pro každý oblouk oválu, dva pro rovinné pasáže, jeden pro rozvětvení nádraží a jeden pro slepou kolej. Díky tomuto bude kolejiště komplexně pokryto.

## 3.2 Vybraný hardware

Po ujasnění výsledné podoby návrhu kolejiště, došlo k výběru vhodného hardwaru. V tomto případě se jednalo o několik rozšiřujících hotových modulů pro platformy Arduino a Raspberry Pi. Tyto mikropočítače budou mít na starosti ovládání klíčových prvků celého kolejiště.

Důležité bylo zejména samotné ovládání kolejiště. Jak již bylo zmíněno, pro tyto účely byla použita platforma Arduino, zejména proto, že se jedná o open-source platformu s velmi podrobnou volně dostupnou dokumentací a s velkým počtem užitečných knihoven.

Arduino obsahuje dostatečné množství pinů, které jsou potřeba pro řízení modelu kolejiště. V neposlední řadě disponuje USB převodníkem použitelným pro programování mikrokontroléru. Konkrétně byla zvolena deska Arduino Mega 2560, která již byla podrobně zmíněna v kapitole 2.6 včetně technických detailů. Programování tohoto mikropočítače bude realizováno v jazyku Wiring, který je podmnožinou jazyka C++.

Pro ovládání výhybek byla vybrána deska obsahující 8 kusů relé. V praxi bude však použito pouze šest relé, neboť kolejiště bude disponovat pouze třemi výhybkami (každá má dvě polohy - jsou potřeba dvě relé pro každou z nich). Pro napájení relé bude nutno vytvořit obvod s napájením 5V.

Pro napájení úseků budou použity tři dvojité H-můstky (jelikož bude šest úseků). K indikaci činnosti H-můstek dojde k přidání indikačních diod (jedna barva bude znamenat napájení pro jeden směr, jiná barva bude znamenat napájení pro opačný směr). Vzhledem k bezpečnosti bude napájení můstek doplněno o ochranný obvod s vratnými pojistkami, které budou chránit H-můstky před zničením. Obvod byl doplněn o LED diody, indikující napájení jednotlivých H-můstek.

Vzhledem k potlačení možného rušení mikroprocesoru byly použity optočleny. Pro každý H-můstek byl použit dvojitý optočlen.

Pro připojení modelů kolejiště byly použity tři konektory, dva pro úseky a jeden pro výhybky. Podrobně o této řídicí desce se všemi komponentami pojednává kapitola 4.4.

Pro účely snímání obrazu a vyhodnocování byl zvolen mikropočítač Raspberry Pi, konkrétně model Model 3 B, který byl podrobně zmíněn v kapitole 2.8. Tento mikropočítač disponuje velkou pamětí a zároveň je dostatečně rychlý, aby tuto činnost vykonával v rozumném čase. Dále disponuje možností vytvoření webového serveru, což bude využito pro ovládání přes Wi-Fi.

Programování tohoto mikropočítače bude realizováno v programovacím jazyce Python, neboť tento jazyk nabízí velké množství knihoven s kvalitně zpracovanou dokumentací a rozsáhlé možnosti zpracování obrazu. Pro vytvoření webového rozhraní bude užít framework Flask.

K tomuto mikropočítači byla zakoupena kamera, která bude fotit kolejiště v určitých časových okamžicích. Byly k ní pořízeny i dva vyměnitelné objektivy s jinými parametry, které lze v případě potřeby vyměnit. Vzhledem k délce flat ribbon kabelu od kamery, bude nutno tento mikropočítač umístit blízko ke kameře.

Mikropočítače Arduino a Raspberry Pi budou vzájemně propojeny USB kabelem, díky kterému bude realizována komunikace přes USB rozhraní. Tento kabel bude zároveň sloužit jako napájení pro Arduino. O vzájemné komunikaci mezi mikropočítači podrobně pojednává kapitola [4.7](#).

## Kapitola 4

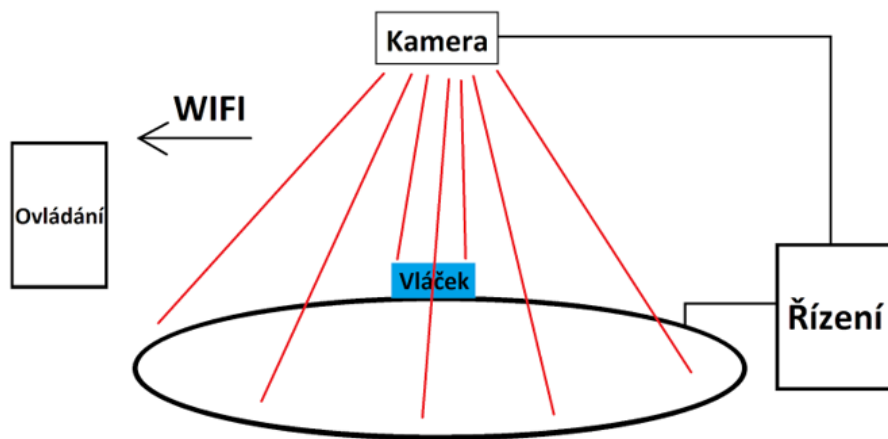
# Návrh a implementace

Tato kapitola detailně popisuje průběh realizace vlastní práce od prvotního návrhu, sestavení, programování jednotlivých algoritmů a závěrečného testování.

### 4.1 Návrh řešení

Finálním výsledkem této práce je plnohodnotné kolejiště ovládané pomocí webového rozhraní, se kterým uživatel komunikuje prostřednictvím Wi-Fi.

Součástí tohoto kolejiště je kamera, která se nachází kolmo nad ním. Snímá pohyb na kolejišti a vyhodnocuje pořízené snímky, které slouží k řízení kolejiště a případnému zabránění srážky vláčků.



Obrázek 4.1: Blokové schéma výsledného řešení.

Bylo nutné také rozvrhnout práci na části, které bude potřeba řešit. Jednalo o hardwarovou a softwarovou část. První z nich má také větší prioritu, neboť bez ní nemá smysl řešit druhou část.

Více tuto dekompozici problémů popisuje následující tabulka:

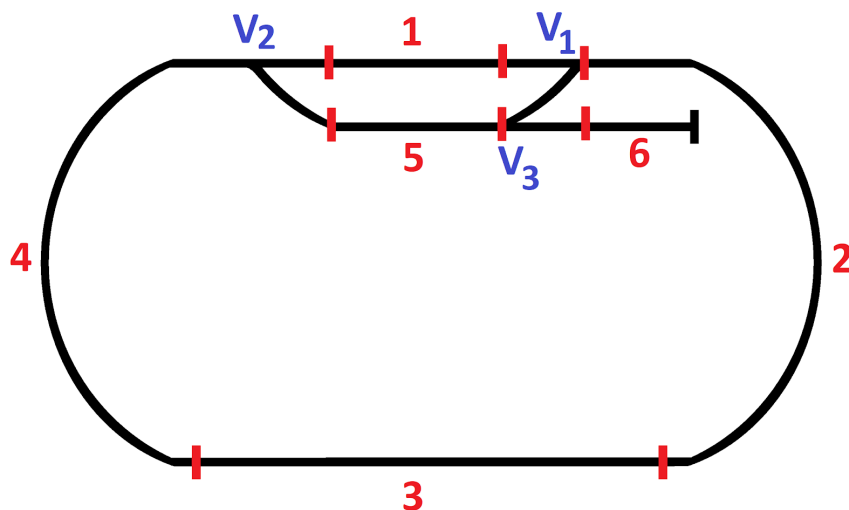
| HW              | SW                |
|-----------------|-------------------|
| Koleje, výhybky | Řízení kolejiště  |
| Vlázky          | Zpracování obrazu |
| Ovládání        | Vyhodnocování     |
| Kamera          | Ovládání          |
| Mikropočítač    | Webový server     |

Tabulka 4.1: Dekompozice problémů.

Po rozdělení práce na jednotlivé části došlo k ucelenému obrazu o celé práci.

## 4.2 Návrh modelového kolejiště

Před samotnou fyzickou realizací bylo nutno si nejprve vytvořit grafický návrh samotného kolejiště. Podle připomínek vedoucího bylo vytvořeno kolejiště, které reflektuje všechny možné situace – výhybky, nádraží a slepou kolej. Konečné rozměry kolejiště v měřítku H0 jsou 120 x 180 cm. Dále také došlo k rozvržení úseků kolejí. Úseků se zde nachází 6 a jsou rovnoměrně a logicky rozmístěny po kolejišti, tak aby se vlaky co nejméně omezovaly.

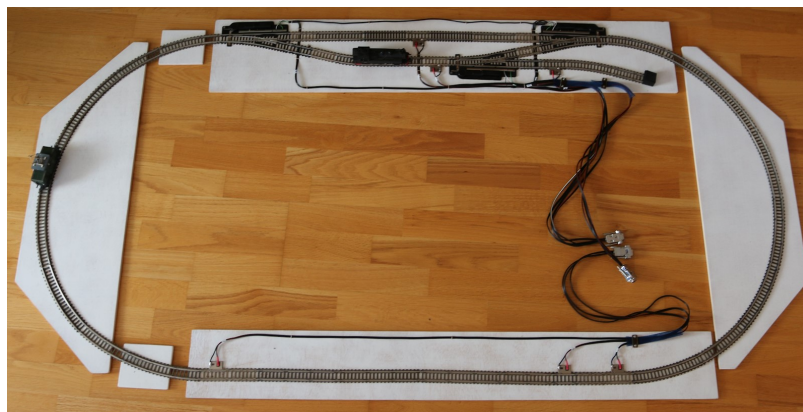


Obrázek 4.2: Grafický návrh uspořádání kolejiště, včetně rozmístění výhybek a úseků.

Jejich počet byl zvolen proto, že Arduino Mega má 12 PWM pinů, které by se daly pro řízení použít (pro každý úsek jsou nutné dva). Rychlost vlaků řídí samotné Arduino.

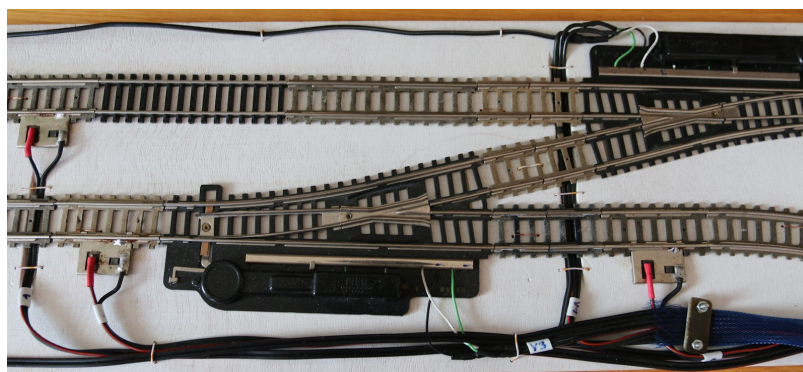
## 4.3 Samotné sestavení kolejiště

Pro jednodušší manipulaci došlo k položení kolejiště na více kusů desek, jelikož jedna deska by byla moc velká a těžká.



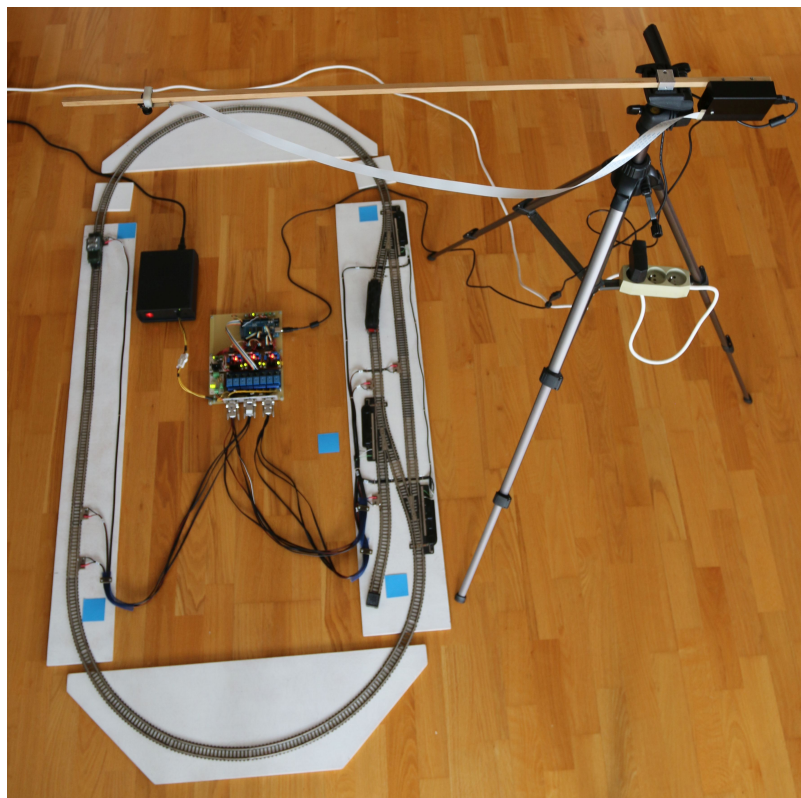
Obrázek 4.3: Výsledné kolejiště.

Dráty pro napájení úseků a výhybek byly popsány a připevněny k deskám. Jelikož v některých částech bylo nutné, aby dráty vedly přes koleje, prvotním řešením bylo postavení můstků ze stavebnice LEGO, nicméně kvůli estetickému dojmu jsem vyfrézovala drážky do desek a tyto dráty byly vedeny pod samotným kolejištěm.



Obrázek 4.4: Vybraný detail kolejiště.

Ke kolejišti byl přidán fotografický stativ, na který je připevněn dlouhý dřevěný modelářský nosník s kamerou a zavěšeným mikropočítačem Raspberry Pi na opačném konci.



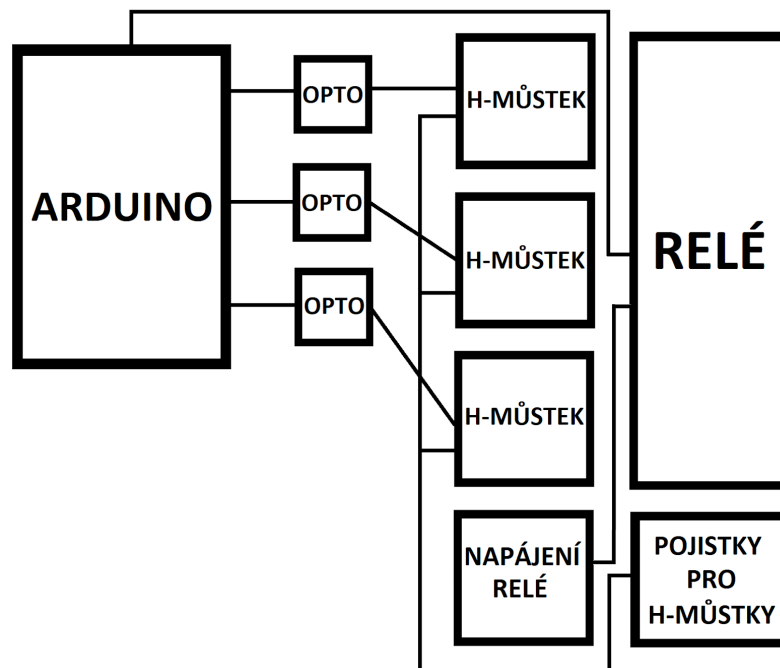
Obrázek 4.5: Fotografie výsledného kolejiště.

Kamera je na stativu umístěna ve výsledné výšce 140 cm nad zemí, která se ukázala jako dostačující pro vyfocení celého kolejiště. Experimentálně byl obraz doostřen otáčením objektivu.

#### 4.4 Řídící deska

Pro ovládání kolejiště byla připravena řídící deska, která byla složena z jednotlivých modulů. Jelikož se jedná o analogové ovládání vláčků, bylo nutno samostatně vyřešit přepínání výhybek a napájení jednotlivých úseků. Všechno tyto úkony zajišťuje naprogramovaný mikrokontrolér.

Nejprve bylo nutno se rozhodnout pro mikrokontrolér, který bude celé kolejiště ovládat. Byla zvolena platforma Arduino, konkrétně model Arduino Mega 2560 (podrobně o něm pojednává podkapitola 2.6). Byla vybrána proto, že je v dnešní době velmi populární a existuje k ní spousta knihoven a kompletní dokumentace. Je také uživatelsky přívětivá, má vlastní vývojové prostředí a dá se programovat v jazyce C/C++. Tento model byl vybrán proto, že má velký výpočetní výkon a také má dostatek pinů (včetně 12 pinů pro PWM) pro ovládání všech potřebných funkcí kolejiště, jednotlivých úseků a výhybek.



Obrázek 4.6: Blokové schéma řídicí desky.

Pro napájení úseků byly použity 3 kusy dvojitých H-můstek L298N. Vzhledem ke zničení jednoho H-můstku při prvotních pokusech, došlo k vytvoření ochranného obvodu obsahující 3 kusy vratných pojistek, aby se tato situace znovu neopakovala. Pro lepší názornost tři LED diody indikují napájení můstek. Tyto vratné pojistky mají tu výhodu, že se nemusí měnit po každém zkratu, ale samy se zotaví z poruchy (po odstranění příčiny).

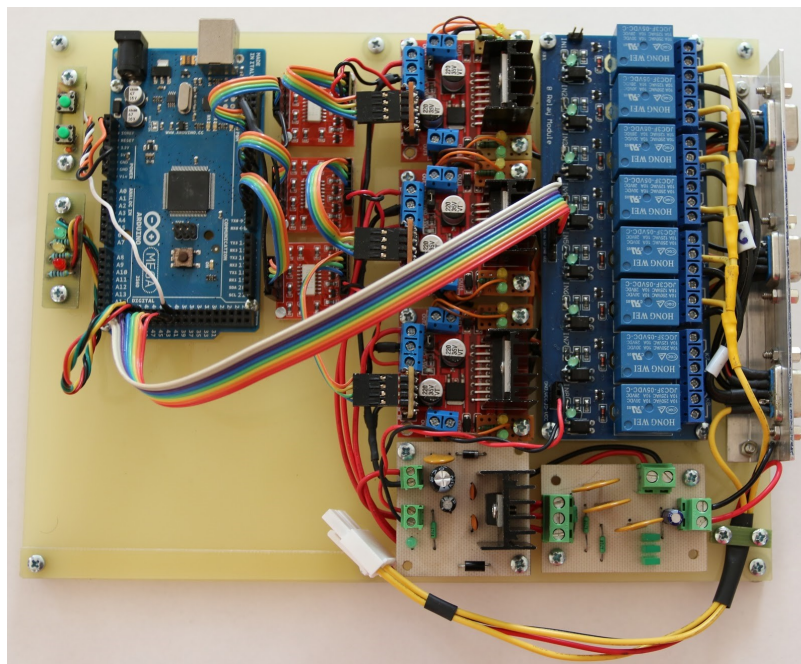
Aby nedocházelo k nežádoucímu rušení Arduina od motorů lokomotiv bylo potřeba použití optického odstínění od H-můstek.

Pro výhybky byla použita deska obsahující 8 kusů relé JQC3F-05VDC-C. Jelikož pro ovládání výhybky je jich nutné pouze šest (pro každou z nich dvě), dvě poslední zbývající relé nebyly použity. Pro napájení relé byl vytvořen elektrický obvod s lineárním stabilizátorem na 5V podle katalogového zapojení.

Pro snazší hledání chyb byly přidány k H-můstkům indikační LED diody. Zelená dioda signalizuje, že je úsek kolejí napájen pro jízdu po směru hodinových ručiček, žlutá znamená proti směru hodinových ručiček.

Byly použity konektory CANON 9 pro snadnější manipulaci (jeden pro výhybky, druhý pro úseky 1, 5, 6 a třetí pro úseky 2, 3, 4.)

Volné místo v levé dolním rohu bylo původně zamýšleno pro mikropočítač Raspberry Pi, ale kvůli délce kabelu ke kameře zde nemohl být umístěn.



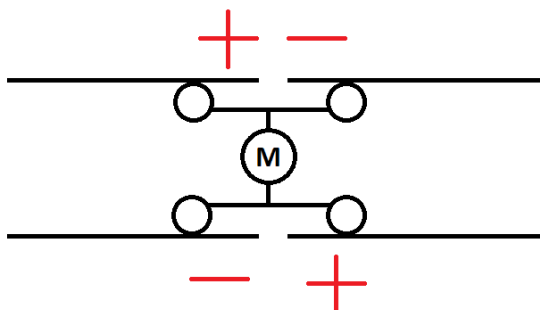
Obrázek 4.7: Výsledný vzhled řídící desky (orientováno stejně jako blokové schéma).

Pro napájení kolejí je použit spínaný zdroj 12V stejnosměrného napětí a pro napájení výhybek transformátor na střídavých 13,5 V.

Výhybky jsou napájeny střídavým napětím, neboť při tomto způsobu napájení mají elektromagnety konstantní sílu. Jednotlivé výhybky jsou přepínány postupně, protože současné přepnutí všech výhybek by zdroj nemusel zvládnout. Protože byly použity staré výhybky, které fungovaly nespolehlivě, byly jim odstraněny koncové spínače a dobu sepnutí řídí samotné Arduino (doba sepnutí byla nastavena experimentálně).

### Ochrana před zničením H-můstků

Kvůli k ochraně H-můstků před zkratem, došlo k hardwarové implementaci bezpečnostního pravidla u Arduina. Toto pravidlo nám říká, že nesmí být dva sousední úseky napájeny s opačnou polaritou, jelikož by mohlo dojít ke zkratu při průjezdu lokomotivy rozhraním úseku a H-můstek by mohl být nenávratně zničen.



Obrázek 4.8: Schématické znázornění zkratu zdroje.

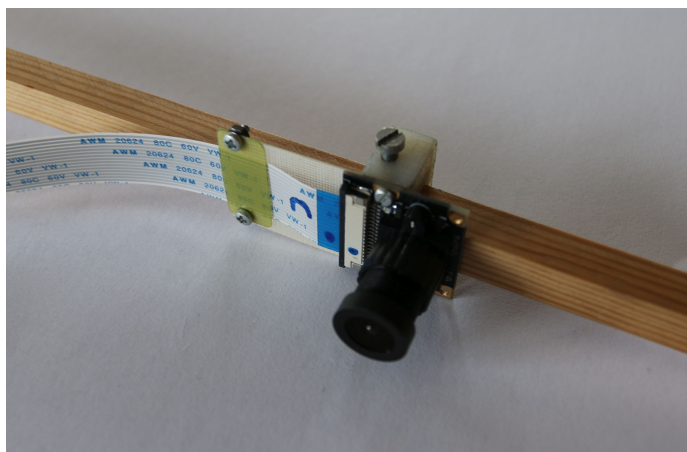


## 4.5 Snímání kamerou

Pro zpracování obrazu byl vybrán mikropočítač Raspberry Pi, protože má oproti Arduino větší paměť (a není nutné načítat na externí SD kartu, což je režie navíc). Je také rychlejší - bylo dosaženo rychlosti snímání fotografie 0,15 s, což je dost podstatný rozdíl (u Arduina se jednalo o čas 0,5 s). Navíc se dá programovat ve více programovacích jazycích. Byl vybrán skriptovací jazyk Python (verze 3.6), pro který existují knihovny pro zpracování obrazu s podrobnou dokumentací (viz kapitola 4.6).

Pro účely snímání obrazu byl koupen modul s kamerou pro Raspberry Pi. Nejedná se o originální Picameru, kvůli finanční dostupnosti a také kvůli tomu, že u ní se nedá měnit objektiv - mohly by pak nastat problémy, že by kamera musela být v příliš velké výšce. Fungují s ní však originální knihovny. K modulu byly dokoupeny dva objektivy s rozdílnou ohniskovou vzdáleností. Na základě výsledných pořízených fotek byl vybrán vhodnější objektiv.

Kamera snímá fotografie v rozlišení 640x480 pixelů. Toto rozlišení se ukázalo jako dostatečné pro rozpoznání vláčků a kalibračních značek. Čím menší rozlišení, tím rychlejší je zpracování obrazu.



Obrázek 4.9: Použitá kamera s upevňovacím modelářským nosníkem.

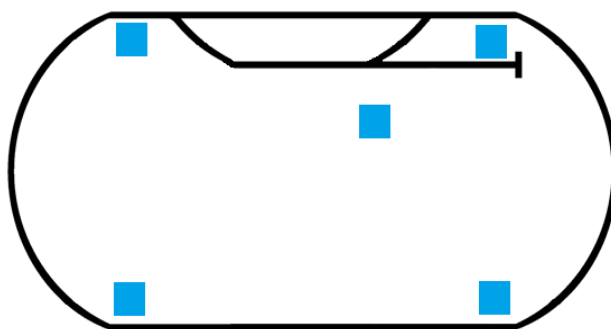
K uchycení byl použit fotografický stativ s dřevěným modelářským nosníkem, na kterém je umístěna kamera s mikropočítačem. Ke kameře byl zakoupen metr dlouhý flat ribbon kabel, aby mohl být mikropočítač ve větší vzdálenosti kvůli lepší stabilitě stativu.

## 4.6 Zpracování obrazu

Pořízené fotografie z kamery bylo potřeba zanalyzovat. Velkou výzvou bylo použití dvou různých lokomotiv, přičemž jedna z nich se pohyboval rychlostí 30 cm/s a druhá 22 cm/s.

Pro potřeby zjištění polohy kolejiště byly přidány kalibrační značky, které určují polohu kolejiště. Čtyři z nich se nachází na obvodu pro nalezení polohy kolejiště, pátá (speciální) se nachází blíže ke středu a slouží pro rozpoznání nádraží a slepé koleje.

Kvůli rozpoznání speciální značky je nutné kolejiště orientovat tak, aby bylo nádraží na spodní straně fotografie.



Obrázek 4.10: Schématické rozložení kalibračních značek na kolejišti.

Tyto značky zároveň slouží na další pomyslné rozdělení úseků, jelikož lokomotivy nesmí najet do stejného úseku, protože by mohla hrozit srážka (pokud by se rychlejší lokomotiva blížila k pomalejší). Dělení podle napájecích úseků nestačí, neboť tyto úseky jsou příliš obrovské.

Na první pohled může toto řešení vypadat trochu nevhodně, ale u reálného modelářského kolejiště by se tyto značky nahradily nějakými modelářskými objekty (například domečky).

Pro zpracování obrazu byla použita knihovna jazyka Python `OpenCV`, která poskytuje nespočet možností práce s obrazy a existuje k ní přehledná a podrobná dokumentace.

## Kalibrační skript

Před prvním použitím je nejprve nutné použít kalibrační skript `calibrate.py`, ve kterém dojde k vyfocení prázdného kolejiště bez vláčků. Poté je nutno fotografii vizuálně zkontrolovat a v případě nevhodně posunutého obrazu ručně přesunout kameru. V případě nespokojenosti s výslednou fotografií, lze přímo ve skriptu nastavit parametry (ISO, čas, expozici, rozlišení,...), neboť ty se později předají provoznímu skriptu pomocí serializačního formátu JSON (z anglického JavaScript Object Notation) a poté již není nutné cokoli měnit.

Dalším důvodem kalibračního skriptu bylo pořízení fotografie prázdného kolejiště, neboť metoda rozpoznávání obrazu pracuje na zjišťování rozdílů dvou po sobě následujících fotek. Bez této fotografie by nebylo na začátku možné poznat, kde se vlaky nachází. Pochopitelně dojde k uložení této fotografie a pozdějšímu použití v provozním skriptu.

V případě spokojenosti s vyfocenou fotografií, již pokračujeme provozním skriptem, který již sleduje celý pohyb na kolejišti.

Druhá možnost kalibrace je u ovládání přes Wi-Fi (více v kapitole 4.8). Stačí použít speciální tlačítko pro vyfocení kolejiště. V tomto případě může dojít k vynechání kalibračního skriptu, pokud je uživatel s kvalitou fotografie spokojen. Pokud nikoli, vrátí se k tomuto skriptu a nastavení kamery si sám upraví.

## Provozní skript

Všechny další úkony vykonává již provozní skript s názvem `app.py`, který běží v nekonečné smyčce. Průběžně fotí nové fotografie a přepisuje ty staré - kvůli omezenému paměťovému prostoru.

## Nalezení značek

Vzhledem k velikosti napájecích úseků došlo k rozdělení na nové podúseky pomocí kalibračních značek. Tyto značky byly rozpoznány podle své unikátní barvy, práh byl zvolen experimentálně při různých světelných podmínkách.

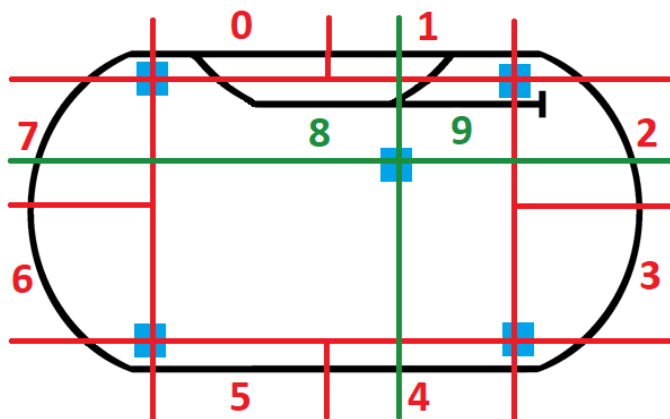


Obrázek 4.11: Nalezení značek a jejich zvýraznění morfologickým filtrem.

Tyto značky jsou dále zpracovány a je vyhodnocen jejich střed. Po této akci je zjištěno 5 souřadnic s jednotlivými body (čtyři značky po obvodu, jedna speciální nejbližší středu kolejiště).

## Rozdělení na podúseky

Nejdříve byly vzaty značky po obvodu a vytvořen z nich polygon (pomocí matematické knihovny `Sympy`). Díky tomu vznikl útvar podobný obdélníku, u kterého poté došlo k rozpůlení jeho stran – kvůli rozdělení na podúseky na obvodu kolejiště. Díky tomu jsou zpřesněné jednotlivé části, ve kterých nesmí být obě lokomotivy zároveň (na stejný napájecí úsek však vjet mohou, nesmí se však k sobě příliš přiblížit - hrozila by srážka).



Obrázek 4.12: Grafické znázornění rozdělených podúseků.

Touto aproximací se přehlíží nepatrnou část kolejiště v rozích, což však nevadí, neboť lokomotiva je zaznamenána těsně před nebo po projetí slepého místa. Tento problém by se dal vyřešit přidáním dalších kalibračních značek.

Speciální značka musí být umístěna v ose s výhybkou  $V_3$ , která se nachází blíže středu kolejiště. Podle této značky se zjišťuje, jestli jsou lokomotivy vlevo nebo vpravo od ní (tím sledujeme, jestli je lokomotiva v boční nádražní koleji a nebo ve slepé koleji.)

Díky všem těmto značkám již došlo k přerozdělení na podúseky, se kterými poté zpracování obrazu dále pracuje. Po předchozích úkonech byly již posbírány všechny informace potřebné k samotnému vyhodnocování obrazu.

### Vyhodnocování polohy

Poloha lokomotiv je určována z rozdílu po sobě následujících fotek. Vzhledem k rozdílné rychlosti lokomotiv, je nutné je od sebe rozlišovat (každá lokomotiva je jinak dlouhá). Lokomotivy od sebe rozeznáme díky změření delší strany nalezeného objektu.

Pro prvotní rozpoznání je díky metodě rozdílu obrazů nutná fotka prázdného kolejiště (toto řeší kalibrační skript, který byl podrobně zmíněn v kapitole 4.6).

V dalších případech již bereme fotografie, které na kterých jsou vyfocené lokomotivy za jízdy. Poloha jednotlivých lokomotiv je poznána podle jejich délky (stejně dlouhé objekty přiřadíme k sobě) a dostaneme jejich novou polohu.

Jednou z variant určování poloh vláčků bylo dělat rozdíly pouze s fotografií prázdného kolejiště. Tento způsob by se však musel doplnit o sledování světelných podmínek (například jako průměr na posledních sto snímcích) a v případě změn reagovat. Proto byl použit způsob, při kterém jsou porovnávány dvě po sobě jdoucí fotografie.

Na obrázku níže s lze vidět 2 různé objekty, což znamená 1 lokomotivu k pohybu. V případě 4 různých objektu by se jednalo o 2 lokomotivy (každá má svou současnou a předchozí polohu). Tvary lokomotivy jsou mírně deformovány, neboť vrhá stín.



Obrázek 4.13: Pohyb jedné lokomotivy po kolejišti – zvýrazněno morfologickým filtrem.

Poté se vypočítají přibližné středy nalezených objektu. Tím přiřadíme ke každé lokomotivě jeden konkrétní bod. Protože podúseky jsou definovány v bodech či v intervalech bodů, začneme tyto hodnoty porovnávat a tím zjišťovat polohu lokomotiv.

Lokomotivy nesmí najet do stejného podúseku. Pokud se to stane, kamera to vyhodnotí jako blížící se srážku a dojde k zastavení vláčků. Vlaky se znova samy nerozjedou, protože pokud by byly na stejném napájecím úseku, nebylo by možné tuto situaci vyřešit. Po zásahu uživatele (posunutím jedné z lokomotiv do jiného úseku), lze vlaky znova uvést do pohybu.

Výsledné demoprogramy jsou vytvořeny tak, aby ukázaly plynulý provoz kolejiště. Více v podkapitole 4.8.

### Podmínka pro počáteční umístění lokomotiv

Lokomotivy lze na koleje umístit libovolně, nesmí však být na stejném podúseku - kamera by to vyhodnotila porušení podmínek a ani by nedošlo k jejich rozjetí. Uživatel si tedy musí dát pozor, aby je umístit na kolejiště korektně.

### Podmínka pro délku vláčků

V případě jízdy s vagóny, je nutné použít maximálně jeden vagónek, kvůli aproximaci objektu na jeden bod. Toto zjednodušení bylo použito kvůli velikosti modelového kolejiště, především kvůli zmenšené velikosti nádraží. Na jeho vnitřní kolej se více než lokomotiva s jedním vagónkem nevejde. V případě větších rozměrů použitých vlaků je nutno provést úpravu a počítat začátek a konec celého objektu. Kvůli délce by také bylo nutno přidat bezpečnostní mezeru, aby se v případě přílišného přiblížení tyto vlaky nesrazily.

## 4.7 Komunikační protokol

Pro účely komunikace mezi mikropočítači došlo k návrhu a implementaci vlastního komunikačního protokolu s názvem ATRAB (z anglického Arduino to Raspberry and Back). Oba mikropočítače spolu komunikují přes USB port, pomocí knihovny `Serial`. Raspberry Pi po vyhodnocení situace posílá Arduino řetězec čísel ve tvaru "XXXXXXYYY". X značí specifikaci chování jednotlivých úseků, Y označuje polohy jednotlivých výhybek.

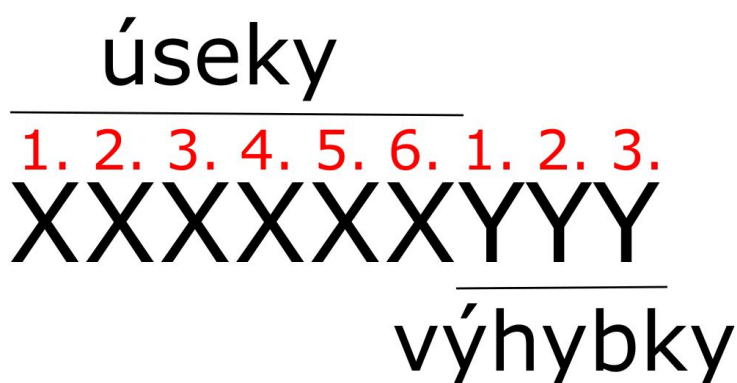
X může nabývat hodnot:

- **0** - úsek není napájen.
- **1** - napájení úseku ve směru pohybu hodinových ručiček.
- **2** - napájení úseku proti směru pohybu hodinových ručiček.

Y může nabývat hodnot:

- **0** - nedělej nic.
- **1** - přepni výhybku do rovného směru.
- **2** - přepni výhybku do odbočky.

Následující obrázek schématicky popisuje zprávy komunikačního protokolu. Prvních šest znaků (znázorněny písmenem X) popisuje napájení úseků, poslední tři znaky (znázorněny písmenem Y) popisují chování výhybek.



Obrázek 4.14: Popis komunikační zprávy, červená čísla značí označení úseků a výhybek.

Například zpráva 001111220, což znamená – první a druhý úsek není napájen, ostatní úseky jsou napájeny, první a druhou výhybku přepne do odbočky a se třetí zůstane v původní poloze.

Arduino po obdržení této zprávy vykoná dané úkony a pošle informaci o přijaté zprávě Raspberry Pi pomocí odeslání jednoho znaku. V případě problému snadněji objevíme řešení.

Typy zpráv co mohou být poslány:

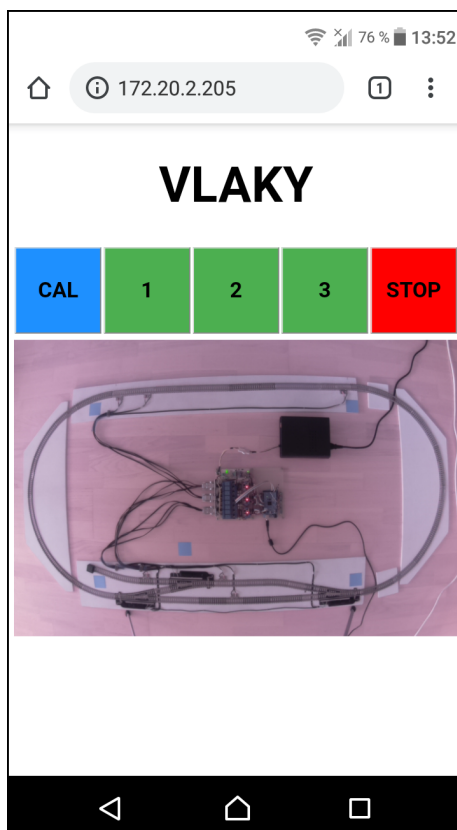
- **1** - příkaz byl v pořádku.
- **2** - příkaz nemá požadovaný formát.

Tento protokol byl navržen tak, pro další možné rozšíření. Jednotlivé číslice mohou nabývat více hodnot, případně se dá dodat více číslic pro ovládání dalších periférií (návěstidla, semaforey, závory,...).

## 4.8 Uživatelské rozhraní

Ovládání bylo realizováno jako webové uživatelské rozhraní. Pro jeho realizaci byl vytvořen webový server na mikropočítači Raspberry Pi, který tuto funkci umožňuje.

Protože tato část nebyla nejdůležitější, bylo navrženo pouze základní rozhraní. Při tvorbě návrhu uživatelského rozhraní byl kladen důraz na jednoduchost a intuitivnost ovládání - aby si vláčky mohly samy ovládat i malé děti.



Obrázek 4.15: Snímek obrazovky uživatelského rozhraní na mobilním telefonu.

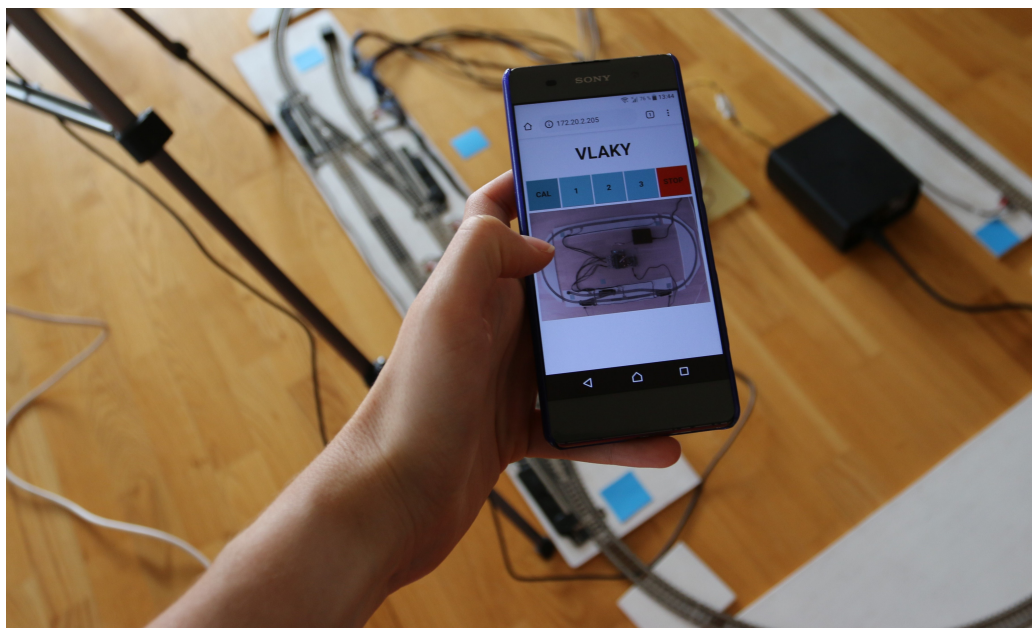
Uživatel nejdříve musí použít tlačítko CAL, které provede kalibraci a zobrazení fotografie aktuální polohy kolejiště (kalibrační fotografie). Pomocí tohoto rozhraní lze rozjet vláčky – byly vytvořeny tři demoprogramy, které běží v nekonečné smyčce.

Bylo přidáno také červené tlačítko STOP, které slouží k okamžitému zastavení lokomotiv.

Označení tlačítek:

- **CAL** – kalibrační tlačítko, vyfotí kolejiště a ukáže danou fotografii. Uživatel se pak rozhodne, jestli je spokojen (v případě nespokojenosti může změnit polohu kamery). V případě nutnosti změnit nějaké konkrétní nastavení (expozici, ISO, rozlišení,...) je nutno změnit v kalibračním skriptu (byl popsán v kapitole 4.6.)
- **1** – spouští první program, ve kterém lokomotivy na sebe čekají na nádraží a střídavě se vydávají na okruh. Až jedna objede okruh a zastaví, vydá se na cestu druhá a tak dále.

- **2** – spouští druhý program, ve kterém je ukázána demonstrace slepého úseku. Lokomotivy na sebe čekají na nádraží a ta, co je u slepé koleje, do ní najede. Následně vycouvá a čeká na nádraží dokud se druhá lokomotiva nevrátí z okruhu.
- **3** – spouští třetí program, ve kterém lokomotivy jezdí okruh střídavě - jedna po směru, poté druhá v protisměru a stále se takto prohazují.
- **STOP** – lokomotivy se ihned zastaví.



Obrázek 4.16: Uživatelské rozhraní v rukou uživatele.

Byla použita technologie HTTP (z anglického Hypertext Transfer Protocol) s CSS (z anglického Cascading Style Sheets) pro vytvoření webového rozhraní. Pro backend byl použit framework Flask, protože je jednoduchý a pro danou problematiku dostačující. Blíže o samotné implementaci pojednává příloha **B**.

## 4.9 Testování

### Testování řídicí desky

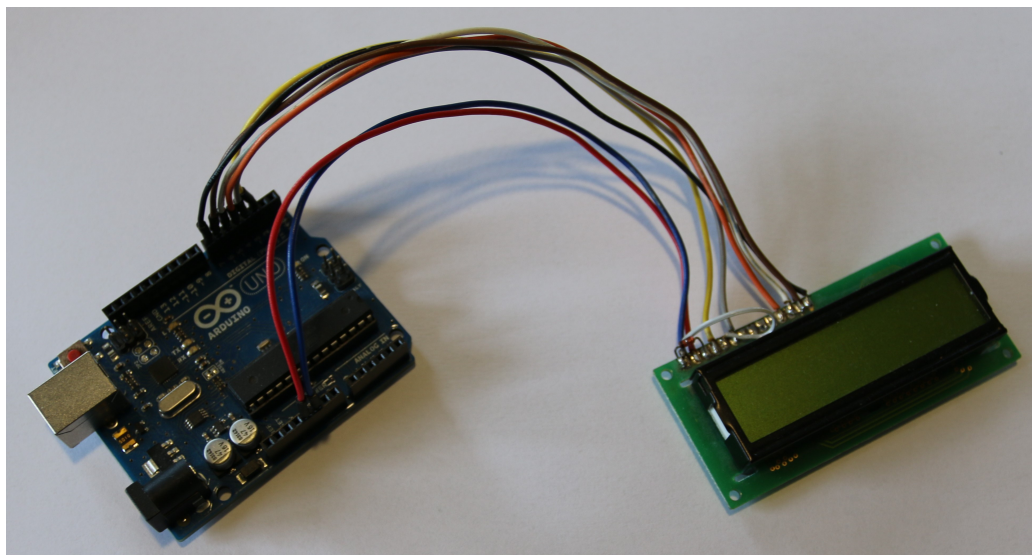
Samotnou řídicí desku bylo nutno pečlivě otestovat, neboť při nekorektním chování by celé kolejiště nemuselo správně fungovat. Bylo nutno zjistit funkčnost H-můstků a relé. Ke každému H-můstku byly přidány dvě indikační LED diody k H-můstkům (pro oba napájecí směry). Modulu s relé již disponoval LED diodami, které se indikovaly funkčnost - při sepnutí dojde ke krátkému rozsvícení příslušné LED diody u daného relé.

Lze tedy přímo sledovat, co se na řídicí desce děje. Byl vytvořen testovací program, který průběžně napájel jednotlivé úseky a postupně přehazoval výhybky. Na desce se to projevovalo úspěšným střídáním rozsvícení správných LED diod.



## Testování komunikačního protokolu

V rámci testování komunikačního protokolu došlo k testování s použitím LCD displeje. Došlo k naprogramování Arduina UNO, které pouze zobrazuje na display informace, které obdrží přes USB port. V případě chyb s nastavením portu, ihned vidíme co Arduino vypisuje a můžeme na to v případě potřeby reagovat.



Obrázek 4.17: Testovací display s deskou Arduino UNO.

Muselo dojít ke shodnému nastavení obou platforem, což znamená:

- Rychlost 9 600 baudů,
- 8 datových bitů,
- 1 stopbit,
- žádná parita.

Díky tomuto došlo k otestování komunikace PC – Arduino, což bylo následně aplikováno na Raspberry Pi – Arduino.

## Testování uživatelského rozhraní

Snadnost a intuitivnost ovládání byla otestována na vzorku třinácti osob ve věku 5 - 78 let. U nikoho z nich se neobjevily potíže s ovládáním a ani se samotnou kalibrací (včetně posunutí samotné kamery).

## Kapitola 5

# Závěr

Cílem této bakalářské práce bylo analyzovat, navrhnout, implementovat a následně demonstrovat ovládání vláčků pomocí mikropočítače. Tento cíl byl splněn.

Byly prostudovány možnosti řízení modelového kolejiště a srovnány jednotlivé způsoby a mikropočítače (včetně jejich vhodnosti pro daný účel).

Poté došlo k navržení ovládání vláčků mikropočítačem. V tomto případě byly využity dvě platformy – Arduino a Raspberry Pi. Arduino ovládá celé řízení kolejiště, napájení úseků a přepínání výhybek. Mikropočítač Raspberry Pi ovládá samotné Arduino, zpracování obrazu a má na starosti ovládání uživatelem.

Pro komunikaci mezi platformami byl navržen vlastní komunikační protokol. Uživatelské rozhraní nabízí možnost kalibrace (vyfocení prázdného kolejiště a zkontrolování, zda-li je celé kolejiště snímáno), spuštění tří nekonečných různorodých programů a zastavení vlaku.

Bylo vymyšleno, realizováno a demonstrováno řízení vláčků pomocí mikropočítačů. Byla realizována komunikace mezi mikropočítači přes USB port pomocí vlastního funkčního komunikačního protokolu. Kamera nad kolejištěm s pomocí kalibračních značek úspěšně poznává polohu vlaku a zabraňuje srážkám.

Možností navázání na tuto práci je spousta. Lze například použít pro řízení vlaků ovládání DCC. Mezi další vylepšení patří například ovládání přes Bluetooth, neboť mikropočítač Raspberry Pi toto umožňuje. Dále by také šlo streamovat provoz na kolejišti online, například na sever Youtube. V neposlední řadě by šlo také zvětšit bezpečnost a více chránit H-můstky před vysokým proudem. Toto například umožňují čidla ACS714, pomocí kterých by byl měřen proud. Také by bylo možno rozpoznávat vlaky podle tvaru s využitím konvolučních sítí.

Přínosem této práce je analýza problematiky řízení modelového kolejiště pomocí mikropočítače, vytvoření funkčního vestavěného systému a realizace komunikace přes USB port mezi platformami Arduino a Raspberry Pi.

# Literatura

- [1] *4. díl o Arduinu – Shieldy* . [Online; navštíveno 06.04.2019].  
URL [https://phgame.cz/PHGame\\_serialy/serialy/zaciname-s-arduinem/4-dil-o-arduinu-shieldy](https://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem/4-dil-o-arduinu-shieldy)
- [2] *Arduino MEGA 2560* . [Online; navštíveno 06.04.2019].  
URL [www.mantech.co.za/datasheets/products/A000047.pdf](http://www.mantech.co.za/datasheets/products/A000047.pdf)
- [3] *Elektrická lokomotiva E499.119 ČSD (H0)* . [Online; navštíveno 30.03.2019].  
URL <http://www.modely.biz/modelova-zeleznice-2/elektricka-lokomotiva-e499-119-csd/>
- [4] *Elektrické zapojení kolejiště* . [Online; navštíveno 08.04.2019].  
URL [http://masinky2003.wz.cz/zapojeni\\_stanice.html](http://masinky2003.wz.cz/zapojeni_stanice.html)
- [5] *NMRA Standards and Recommended Practices* . [Online; navštíveno 11.04.2019].  
URL <https://nmra.org/index-nmra-standards-and-recommended-practices>
- [6] *N Scale Shinkansen (NEW)* . [Online; navštíveno 30.03.2019].  
URL <https://www.modeltrainplus.net/collections/n-scale-shinkansen-new>
- [7] *Pro začátečníky*. [Online; navštíveno 27.03.2019].  
URL <https://www.elektrickevlacky.cz/modelova-zeleznice-zacatecnici.php>
- [8] *NMRA STANDARD* . Červenec 2004, [Online; navštíveno 11.04.2019].  
URL <https://www.nmra.org/sites/default/files/s-92-2004-07.pdf>
- [9] *Modelářství*. Březen 2008, [Online; navštíveno 28.03.2019].  
URL <https://cs.wikipedia.org/wiki/Model%C3%A1%C5%99stv%C3%AD>
- [10] *DCC (Digital Command Control)*. Leden 2014, [Online; navštíveno 28.03.2019].  
URL <http://www.vlacky.com/dcc/>
- [11] *DCC pro laiky a začátečníky* . Duben 2016, [Online; navštíveno 11.04.2019].  
URL <http://www.zavavov.cz/cz/modelova-zeleznice/elektronika/89-dcc-pro-laiky-a-zacatecniky/>
- [12] *What is DCC Plus Plus* . Únor 2016, [Online; navštíveno 12.04.2019].  
URL <https://github.com/DccPlusPlus/BaseStation/wiki/What-is-DCC--Plus-Plus>
- [13] *Pulzně šířková modulace* . Prosinec 2018, [Online; navštíveno 16.04.2019].  
URL [https://cs.wikipedia.org/wiki/Pulzn%C4%9B\\_%C5%A1%C3%AD%C5%99kov%C3%A1\\_modulace](https://cs.wikipedia.org/wiki/Pulzn%C4%9B_%C5%A1%C3%AD%C5%99kov%C3%A1_modulace)

- [14] *Arduino*. Březen 2019, [Online; navštíveno 27.03.2019].  
URL <https://cs.wikipedia.org/wiki/Arduino>
- [15] *Arduino IDE for Visual Studio*. 2019, [Online; navštíveno 30.03.2019].  
URL <https://marketplace.visualstudio.com/items?itemName=VisualMicro.ArduinoIDEforVisualStudio>
- [16] *Arduino Mega 2560 Rev3*. 2019, [Online; navštíveno 06.04.2019].  
URL <https://store.arduino.cc/mega-2560-r3>
- [17] *Arduino Products*. 2019, [Online; navštíveno 31.03.2019].  
URL <https://www.arduino.cc/en/Main/Products>
- [18] *Compare board specs*. 2019, [Online; navštíveno 31.03.2019].  
URL <https://www.arduino.cc/en/Products/Compare>
- [19] *English WINTRACK version*. 2019, [Online; navštíveno 30.03.2019].  
URL <https://www.wintrack.de/>
- [20] *FITkit*. Březen 2019, [Online; navštíveno 08.04.2019].  
URL <https://cs.wikipedia.org/wiki/FITkit>
- [21] *Mikrokontrolér PIC*. Únor 2019, [Online; navštíveno 08.04.2019].  
URL [https://cs.wikipedia.org/wiki/Mikrokontrol%C3%A9r\\_PIC](https://cs.wikipedia.org/wiki/Mikrokontrol%C3%A9r_PIC)
- [22] *PWM*. 2019, [Online; navštíveno 16.04.2019].  
URL <https://www.arduino.cc/en/Tutorial/PWM>
- [23] *Raspberry Pi*. Březen 2019, [Online; navštíveno 06.04.2019].  
URL [https://cs.wikipedia.org/wiki/Raspberry\\_Pi](https://cs.wikipedia.org/wiki/Raspberry_Pi)
- [24] *Raspberry Pi 3 Model B+ 64-bit 1GB RAM*. 2019, [Online; navštíveno 08.04.2019].  
URL <http://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit.html>
- [25] *Simple Computer Aided Railway Modeller*. 2019, [Online; navštíveno 30.03.2019].  
URL <https://www.scarm.info/index.php>
- [26] *What is Arduino?*. 2019, [Online; navštíveno 30.03.2019].  
URL <https://www.arduino.cc/en/Guide/Introduction>
- [27] Benchoff, B.: *Introducing the Raspberry Pi 3*. Únor 2016, [Online; navštíveno 07.04.2019].  
URL <https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>
- [28] Bohouš: *Software pro projektování a kreslení*. Září 2012, [Online; navštíveno 30.03.2019].  
URL <http://www.masinky.info/2012/09/software-pro-kresleni/>
- [29] Hankovec, D.: *Pulzně šířková modulace*. [Online; navštíveno 16.04.2019].  
URL <http://www.dhservis.cz/psm.htm>

- [30] Justo, P. D.: *Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board* . Prosinec 2015, [Online; navštíveno 07.04.2019].  
URL <https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/>
- [31] Mahel, I.: *Měřítka a velikosti železničních modelů*. [Online; navštíveno 27.03.2019].  
URL <https://www.parostroj.net/modely/meritka/meritka.htm>
- [32] Okoi, M. D.: *10 Operating Systems You Can Run on Raspberry Pi in 2019* . Březen 2019, [Online; navštíveno 06.04.2019].  
URL <https://www.fossmint.com/operating-systems-for-raspberry-pi/>
- [33] Raj, A.: *What is the difference between Arduino mega, Arduino mega 1280 and Arduino mega 2560?* . Červenec 2016, [Online; navštíveno 06.04.2019].  
URL <https://www.quora.com/What-is-the-difference-between-Arduino-mega-Arduino-mega-1280-and-Arduino-mega-2560>
- [34] Tom, I.: *Wemos ESP8266 Getting Started Guide With Arduino IDE* . [Online; navštíveno 06.04.2019].  
URL <https://www.instructables.com/id/Wemos-ESP8266-Getting-Started-Guide-Wemos-101/>
- [35] Trávníček, S.: *[PIC programování] – Úvod* . Duben 2016, [Online; navštíveno 08.04.2019].  
URL <https://www.allcomp.cz/blog/?p=233>
- [36] Voda, Z.: *Seznámení s Arduinem* . Říjen 2014, [Online; navštíveno 06.04.2019].  
URL <https://arduino.cz/seznameni-s-arduinem/>
- [37] Voda, Z.: *Arduino klony – Co to je? Koupit či nekoupit?* . Červen 2017, [Online; navštíveno 06.04.2019].  
URL <https://arduino.cz/arduino-klony-co-to-je-koupit-ci-nekoupit/>
- [38] Voda, Z.: *Průvodce světem Arduina*. SparkFun, 2017, ISBN 978-80-87106-93-8.

# Příloha A

## Obsah přiloženého CD

Na přiloženém disku se nachází:

- `/calibrate/calibrate.py` - kalibrační skript,
- `/calibrate/setting.json` - nastavení kamery ve formátu JSON,
- `/control/control.ino` - program pro řízení Arduina,
- `/web-server/templates/main.html` - webové rozhraní,
- `/web-server/static/first.bmp` - kalibrační fotografie,
- `/web-server/static/actual.bmp` - aktuální fotografie,
- `/web-server/app.py` - hlavní program pro Raspberry Pi,
- `/web-server/find_trains/__init__.py`
- `/web-server/find_trains/controller.py` - detekce podúseků,
- `/web-server/find_trains/marker_detector.py` - nalezení kalibračních značek,
- `/web-server/find_trains/train_detector.py` - detekce vláčků,
- `/test/test.cpp` - program pro testování řídicí desky,
- `xkavan05.pdf` - text této bakalářské práce ve formátu PDF,
- `/sablona` - složka se zdrojovými kódy k textu této práce, šablona je doplněna o vlastní text a obrázky.
- `Readme.txt` - základní informace.

## Příloha B

# Implementace software

Tato část přílohy pojednává o programové části vytvořené pro účely této bakalářské práce. První částí je ovládání kolejiště pomocí Arduina, druhou částí je rozpoznávání obrazu na mikropočítači Raspberry Pi, třetí částí je implementace navrženého komunikačního protokolu a čtvrtou částí je tvorba uživatelského rozhraní pro ovládání kolejiště přes Wi-Fi.

Do souboru `/etc/rc.local` byly přidány odkazy na všechny programy, díky tomu se spustí již při nastartování mikropočítače Raspberry Pi a uživatel ovládá kolejiště pouze pomocí mobilu (výjimku tvoří pouze situace, kdy bude chtít uživatel zasáhnout do nastavení parametrů kamery - viz kapitola 4.6.)

### Implementace ovládání kolejiště

Ovládání kolejiště bylo realizováno v programovacím jazyce Wiring, který slouží přímo pro programování platformy Arduino. Veškerý kód se nachází v souboru `control.ino`.

Pro definování pinů úseků, výhybek a kontrolních diod byla vytvořena makra, aby v případě změny pinů stačil nepatrný zásah do začátku kódu.

Funkce `Section()` má řídit napájení úseků. V případě náhlé změny polarity kolejí, vkládá pauzu. Jinak by hrozilo zničení lokomotiv.

Funkce `Switch()` přepíná výhybky. K tomuto dochází postupně podle priorit výhybek, neboť současné přepnutí by zdroj nemusel zvládnout. Pauza mezi přepínáním byla experimentálně nastavena na 75ms.

Funkce `CheckCommand()` má na starost ochranu H-můstků. Hlídá aby nebyly sousední úseky napájeny s opačnou polaritou, hrozilo by nenávratné zničení H-můstků. Dochází ke kontrolám všem možným kombinacím.

Arduino prvně načte sekvenci znaků, kterou obdrží přes USB port. Zkontroluje, jestli je to validní hodnota v intervalu 48 – 50 v ASCII (z anglického American Standard Code for Information Interchange) tabulce (odpovídá číslicím 0,1,2). Poté v případě validního příkazu vykoná požadované akce.

## Implementace rozpoznávání obrazu

Rozpoznávání obrazu bylo realizováno v programovacím jazyce Python. Implementace je postavena na skriptu `app.py`, který používá modul `find_trains`. Byly použity primárně knihovny `Opencv`, `Sympy` a `Scikit-image`.

Skript `marker_detector.py` je postaven na třídě `MarkerDetector`, která ve své metodě `detect()` hledá kalibrační značky, které byly zadány v HSV (z anglického (Hue, Saturation, Value) barevném modelu).

Skript `controller.py` obsahuje třídu `Controller`, která pracuje se souřadnicemi kalibračních značek. Pomocí funkce `centroid()` dojde k rozpůlení vzdálenosti mezi obvodovými značkami (a tím i k rozdělení podúseků). V metodě `_detect_segment` přiřazuje podle jednotlivých souřadnic dané podúseky a vrací jejich indexy.

Skript `train_detector.py` hledá rozdíly na dvou po sobě jdoucích fotografiích. Experimentálně došlo k nastavení prahu, který určuje, zda-li se jedná o změnu či nikoli (bylo nutné vzít v úvahu jevy jako jsou například změny světla v místnosti.)

Všechny tyto skripty jsou volány z hlavního souboru `app.py`. Ve kterém nejprve dojde k načtení kalibrační fotografie pomocí funkce `imread()`. Z této fotografie jsou následně zjištěny polohy kalibračních značek. Poté jsou v nekonečné smyčce `while True:` foceny nové fotografie a tím cyklicky přepisovány staré (kvůli omezenému paměťovému prostoru.)

## Implementace komunikačního protokolu

Navržený protokol byl implementován u obou mikropočítačů, které spolu navzájem kooperují.

V případě Arduina, dochází ke čtení sekvence 10 znaků (9 znaků pro specifikaci činnosti, poslední znak je `\n`) přijatých přes USB port pomocí funkce `Serial.read()`. Arduino posílá pomocí příkazu `Serial.print()` zprávu Raspberry Pi. Typy zpráv co mohou být poslány jsou číslíky 1 (příkaz byl v pořádku) nebo 2 (příkaz nemá požadovaný formát - nemůže být vykonán). V případě odeslání hodnoty 1 vykoná požadované úkony.

Raspberry Pi posílá sekvence znaků Arduinu a naopak čte zprávy od něj (jestli se přenos podařil či nikoli). Byl využit modul `pySerial` programovacího jazyka Python. Jelikož má Raspberry Pi 4 USB porty, došlo ke konkrétnímu použití portu `/dev/ttyACM0` pro vzájemnou komunikaci.

## Implementace webového rozhraní

Webové rozhraní bylo realizováno prostřednictvím frameworku `Flask`. Celá implementace je postavena na skriptu `app.py`, ve kterém jsou definovány jednotlivé akce po stisku tlačítek a samotné spuštění webového serveru.

Frontend se nachází ve složce `/templates` v souboru `main.html`, ve kterém je zároveň zahrnut CSS skript pro vizuální úpravu (barva tlačítek, písma,...). Samotná načtená fotografie je uložena ve složce `/static`, jelikož `Flask` k ní přistupuje jako ke statickému objektu.

Fotografie je se objeví po zmáčknutí tlačítka `CAL`, jak již bylo zmíněno v podkapitole o kalibračním skriptu. Také došlo k vytvoření formuláře s pěti tlačítky typu `submit`. Každé tlačítko vykonává jiný typ demoprogramu.