

DELTA – Střední škola informatiky a ekonomie s.r.o

Ke Kamenci 151, 530 03 Pardubice

**DELTA**

## **Maturitní projekt**

Ozodráha

Jméno a příjmení: Matěj Beran

Třída: 4.A

Studijní obor: Informační technologie 18-20-M/01

Školní rok: 2022/2023

# Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Matěj Beran*  
Pro školní rok: *2022/2023*  
Třída: *4. A*  
Obor: *Informační technologie 18-20-M/01*

Téma práce: *Ozodráha – miniaturní roboti řízení pomocí LED*  
Vedoucí práce: *RNDr. Jan Koupil, Ph.D.*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem práce je ověřit možnost řídit miniaturní roboty Ozoboty pomocí LED diody a navrhnout hardware a software pro ovládání dráhy. Realizace bude zahrnovat jak práci s jednočipovým mikrokontrolerem, tak jeho řízení z desktopové a případně mobilní aplikace (mobilní aplikace je uvedena jako rozšiřující možnost, nikoli požadavek).

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- *Září–říjen*: Analýza existujících řešení problematiky. Seznámení s Ozobotem, mikrokontrolerem, ověření možnosti řízení pomocí RGB diody v kontinuálním režimu. Průzkum možností pro rozšíření počtu výstupních portů (de/multiplexor, port expander, shift register, SPI LED drivery) a kontroly obsazenosti dráhy (optočleny a jiné možnosti).
- *Listopad*: Návrh aplikace pro mikrokontroler a komunikace mezi mikrokontrolerem a desktopem s ohledem na případnou komunikaci pomocí bezdrátové technologie (Bluetooth).
- *Prosinec*: Návrh GUI pro řízení dráhy z desktopové aplikace
- *Leden*: Další práce na realizaci ozodráhy a řídicího software
- *Únor*: Debugging, případně vývoj mobilní aplikace
- *Březen*: Dokumentace maturitního projektu

## Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze zdroje uvedené v seznamu bibliografických záznamů.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 31. 3. 2023 .....

Matěj Beran

## **Poděkování**

Chtěl bych poděkovat RNDr. Janu Koupilovi, Ph.D. za vedení projektu, a za podnětné návrhy na vývoj a tvorbu projektu i této práce. Dále bych rád poděkoval všem, kdo přispěli svým názorem k návrhu a tvorbě tohoto projektu.

## **Abstrakt**

Tato práce dokumentuje návrh, vývoj a tvorbu modulární stavebnice drah pro roboty Ozobot, zaměřenou na výuku základů programování u dětí školního věku. Realizace projektu zahrnuje fyzickou část, elektrotechnickou část a softwarovou/firmwarovou část.

## **Klíčová slova**

Programování, Elektrotechnika, Ozobot, C#, výuková pomůcka

## **Abstract**

This document describes the design, development and creation of a modular building set of tracks for Ozobot robots, aimed to improve the education of basic programming principles of school-aged children. The implementation of this project includes the physical part, electrical engineering part and software/firmware part.

## **Keywords**

Programming, Electrical Engineering, Ozobot, C#, educational tool



# Obsah

<b>1 Úvod</b>	<b>9</b>
1.1 Ozobot a jeho svět . . . . .	9
1.2 Existující řešení . . . . .	10
<b>2 Návrh projektu</b>	<b>11</b>
2.1 Cíl projektu . . . . .	11
2.2 Architektura . . . . .	11
2.3 Řídící aplikace . . . . .	13
2.3.1 Využití technologie . . . . .	13
2.3.2 Komunikace s ovládacím modulem . . . . .	14
2.3.3 Přijímání dat od uživatele . . . . .	14
2.4 Ovládací modul . . . . .	17
2.4.1 Fyzické provedení . . . . .	17
2.4.2 Software . . . . .	18
2.5 Podřízený modul . . . . .	19
2.5.1 Fyzické provedení . . . . .	19
2.5.2 Software . . . . .	21
2.6 Pasivní prvky . . . . .	21
2.7 Komunikace . . . . .	22
2.7.1 Komunikace mezi aplikací a ovládacím modulem . . . . .	22
2.7.2 Komunikace mezi ovládacím a podřízenými moduly . . . . .	23
<b>3 Závěr</b>	<b>25</b>





# Kapitola 1

## Úvod

### 1.1 Ozobot a jeho svět

Ozobot je komerčně dostupný, zhruba 2,5 na 2,5 centimetru velký robot, určený pro výuku základních principů programování. Robot se pohybuje díky dvěma malým elektromotorům, ovládajících každý jedno kolo robota. Elektromotory jsou řízeny deskou s mikrokontrolérem, který kromě elektromotorů ovládá také LED diodu sloužící pro vizuální odezvu uživateli. Jediným interakčním prvkem je tlačítko na boku, sloužící k zapnutí a vypnutí robota.

Ozobot je robot typu „black line follower“ - pokud Ozobot stojí na černé čáře o tloušťce minimálně 5 milimetrů, automaticky po ní pojede. K této funkci Ozobot využívá pět senzorů, které rozeznávají černou a bílou barvu. Pokud Ozobot nejede prostředním senzorem po černé čáře, ale po některém z bočních senzorů, Ozobot upraví svůj kurz tak, aby jel opět prostředním senzorem po čáře.

Ozobot má navíc funkci patentovanou firmou Ozobot, která je klíčem k tomuto projektu - funkce Color Codes. Tato funkce umožňuje Ozobotu upravovat svoje chování, tedy jeho rychlost, směr jízdy, nastavení časovače a podmínek pro ”výhru”. Těsně před prostředním senzorem čáry je umístěn i se senzor barev citlivý především na rozlišení červené, zelené a modré. Pokud tedy Ozobot přejede přes kombinaci těchto tří barev v tří až čtyřčlenné

sekvenci, robot upraví svoje chování podle dané sekvence (viz obr. 1.1 Tabulka sekvencí).

Firma Ozobot nabízí způsob použití těchto sekvencí dvěma způsoby - Barevnými fixy, nebo samolepkami, obojí je dostupné od řečené firmy k zakoupení.



Obrázek 1.1: Tabulka sekvencí.

## 1.2 Existující řešení

Ozobot je uzpůsoben jízdě po papíru, a podle toho tudíž existují předem zmiňované způsoby aplikace barevných sekvencí, tedy barevnými fixy a samolepkami. Krom tohoto standardního způsobu existuje alternativa, a to v podobě tabletu s nainstalovanou aplikací. Ta může být přímo určená pro platformu Ozobot, např. Ozogroove, stačí ovšem i obyčejná aplikace typu „plátno“, např. Sketchbook, nebo Infinite Painter.

# Kapitola 2

## Návrh projektu

### 2.1 Cíl projektu

Hlavním cílem tohoto projektu je vytvořit modulární a intuitivní stavebnici, která po sestavení umožňuje Ozobotu, nebo Ozobotům jezdit po sestavené dráze.

Snaží se vyřešit fundamentální problém klasického způsobu použití Ozobotů - absolutní statika klasické dráhy po jejím sestavení. Pokud dráhu nakreslíme na papír, stane se statickou, a s žádnými již nakreslenými prvky nemůžeme hýbat. Papír také představuje vedlejší problémy, mezi nimi např. vpití barev při použití fix, po kterém se barvená sekvence může stát pro Ozobota nečitelnou, nebo nekonzistentnost samolepek, na kterých se může Ozobot zaseknout.

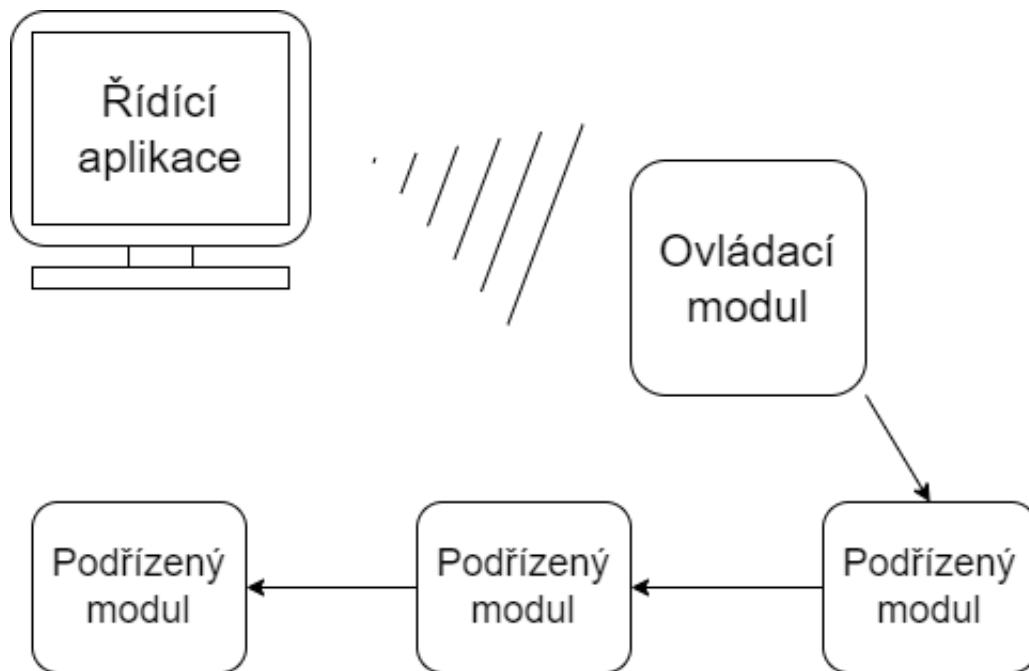
Zároveň se projekt snaží vyřešit problém alternativního přístupu ve formě tabletu, kterým je nedostatek prostoru, zkrátka plocha standardního tabletu je příliš malá na jakékoli normální užívání Ozobota.

### 2.2 Architektura

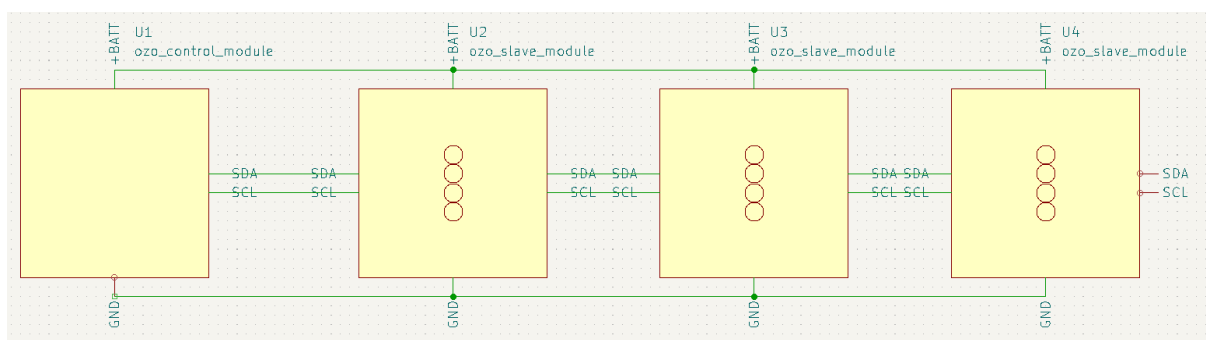
Projekt je rozdělen na tři hlavní a jednu vedlejší část: Řídící aplikaci, Ovládací modul, podřízený modul a pasivní prvky. Každá z těchto částí je nezbytná pro správné fungování

dráhy. Projekt je tak sestaven hierarchicky, tedy Řídící aplikace řídí Ovládací modul, který následovně ovládá podřízené moduly (viz obr. 2.1 Schéma architektury.), pro doplnění dráhy jsou použity pasivní prvky.

Dráha z této stavebnice se skládá na LEGO® desku, a umožňuje tak nekonečné rozšiřování dráhy. Jednotlivé moduly mají vždy dva konektory, aby se k danému modulu dal vždy připojit další. Tyto konektory jsou vzájemně zaměnitelné, tedy „vstupní“ se dá použít jako „výstupní“ a obráceně (viz obr. 2.2 Schéma sestavené dráhy.).



Obrázek 2.1: Schéma architektury.



Obrázek 2.2: Schéma sestavené dráhy.

## 2.3 Řídící aplikace

Řídící aplikace slouží ke komunikaci uživatele s celou dráhou, je klíčovým bodem v celkové architektuře. Aplikace je naprogramována ve frameworku .NET, v jazyce C#, v prostředí WPF.

Aplikace má na starost dvě věci, a to komunikaci s ovládacím modulem, a přijímání dat od uživatele.

### 2.3.1 Využité technologie

#### C#

C#(vyslovuje se "See Sharp") je objektově orientovaný, bezpečně typovaný jazyk vyvinutý společností Microsoft. Patří do rodiny jazyků C a nejvíc se mu podobají jazyky C++, Java a Javascript. Jazyk je součástí .NET Framework a používá se k vývoji různých typů aplikací, jako jsou desktopové aplikace, webové aplikace a mobilní aplikace pro platformy Windows, iOS a Android[1].

#### .NET

.NET je softwarový framework vyvinutý společností Microsoft pro vývoj aplikací na platformě Windows a dalších systémech. Framework zahrnuje soubor nástrojů, knihoven a jazyků pro vývoj, testování a nasazení různých typů aplikací, včetně desktopových aplikací, webových aplikací, mobilních aplikací a her. .NET umožňuje vývojářům psát kód

v jazyce C#, F# nebo Visual Basic a kompilovat ho do Common Language Runtime (CLR), který běží na operačním systému Windows nebo jiných platformách jako Linux a macOS. Díky .NETu jsou aplikace vytvořené v těchto jazycích škálovatelné, bezpečné a snadno spravovatelné[2].

## WPF

WPF (Windows Presentation Foundation) je součástí .NET Frameworku, který umožňuje vývojářům vytvářet bohaté a interaktivní uživatelské rozhraní pro aplikace běžící na platformě Windows. WPF používá deklarativní jazyk XAML (eXtensible Application Markup Language), který odděluje grafické prvky a prezentaci od kódu aplikace, což umožňuje vývojářům snadněji vytvářet a upravovat uživatelské rozhraní.

WPF poskytuje řadu pokročilých funkcionálních prvků, jako jsou animace, přechody, efekty, multi-touch ovládání a podpora pro 2D a 3D grafiku, což umožňuje vytvářet moderní a dynamické uživatelské rozhraní. WPF také poskytuje flexibilní a konfigurovatelné datové vazby, které umožňují vývojářům snadno propojovat datové zdroje s uživatelským rozhraním[3].

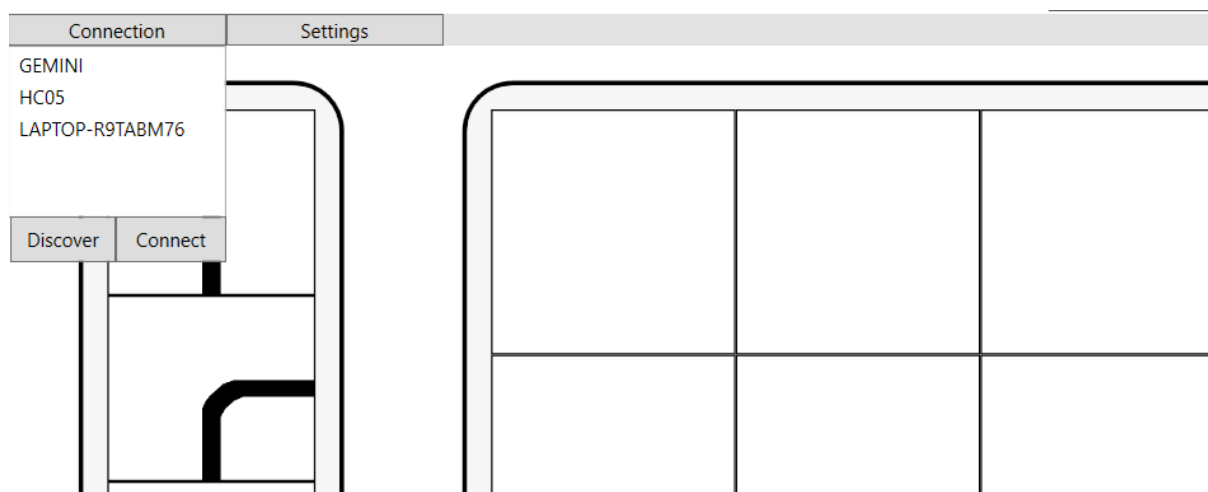
### 2.3.2 Komunikace s ovládacím modulem

Pro komunikaci je ve vizuálním prostředí aplikace určena záložka „Connection“ (viz obr. 2.3 Záložka „Connection“). V ní můžeme hledat Bluetooth zařízení v okolí, a následně se k nim připojit. Aby aplikace mohla používat Bluetooth modul zabudovaný do počítače, využívá balíček 32Feet.

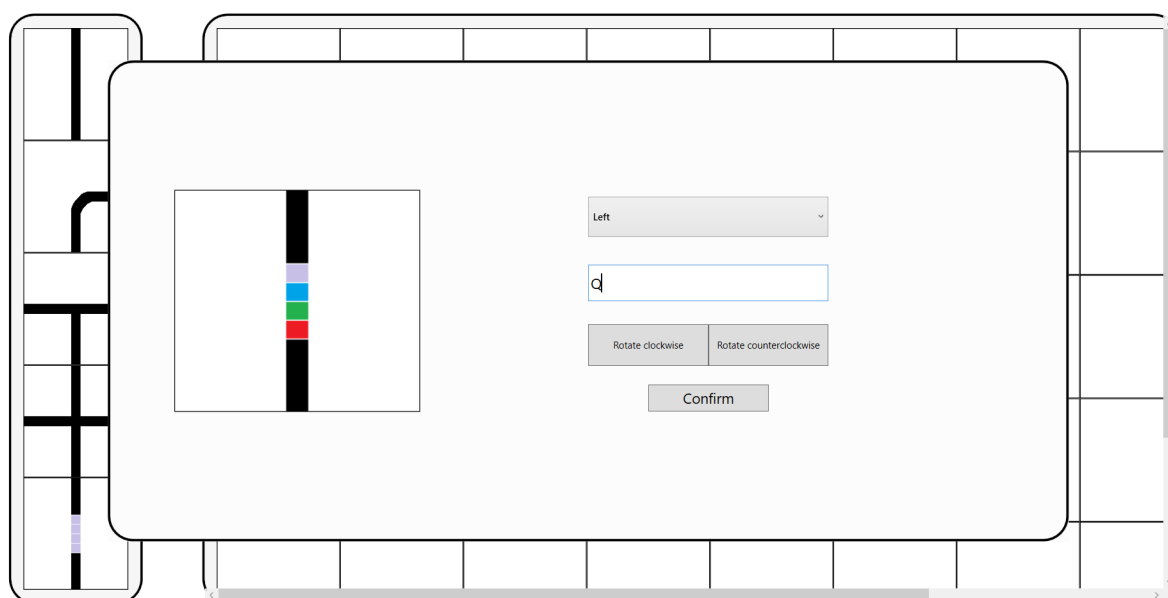
Po navázání komunikace může aplikace vysílat data ovládacímu modulu. Provádí tak vždy, když uživatel změní vlastnosti některého z podřízených modulů ve vizuální reprezentaci dráhy (viz Řídící aplikace - Přijímání dat od uživatele).

### 2.3.3 Přijímání dat od uživatele

Hlavní částí aplikace je vizuální reprezentace dráhy. Jedná se o pole, které může uživatel upravovat tím, že přidává a odebírá jednotlivé bloky dráhy (viz obr. 2.5 Náhled apli-



Obrázek 2.3: Zložka „Connection“.

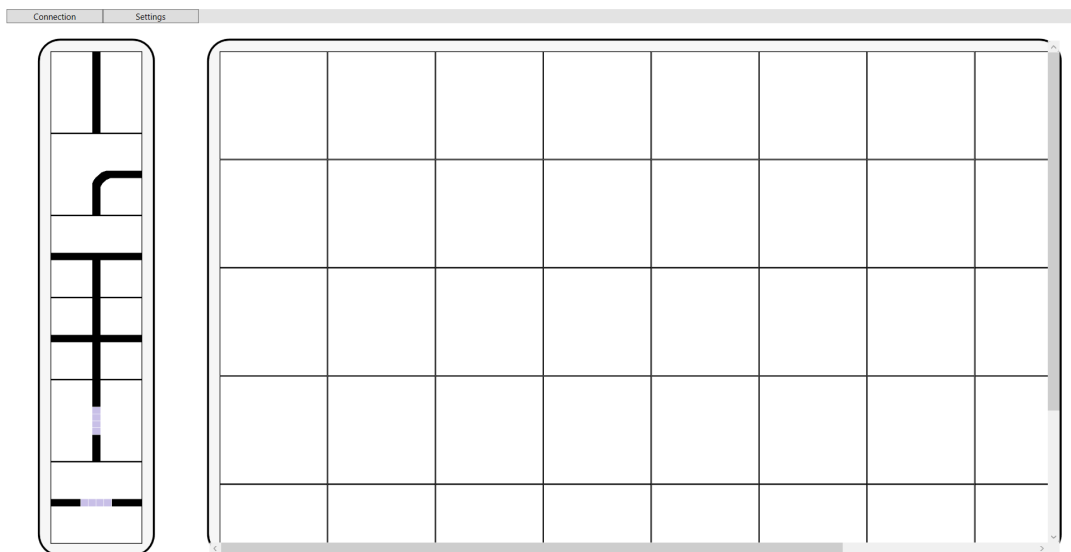


Obrázek 2.4: Přijímání dat od uživatele.

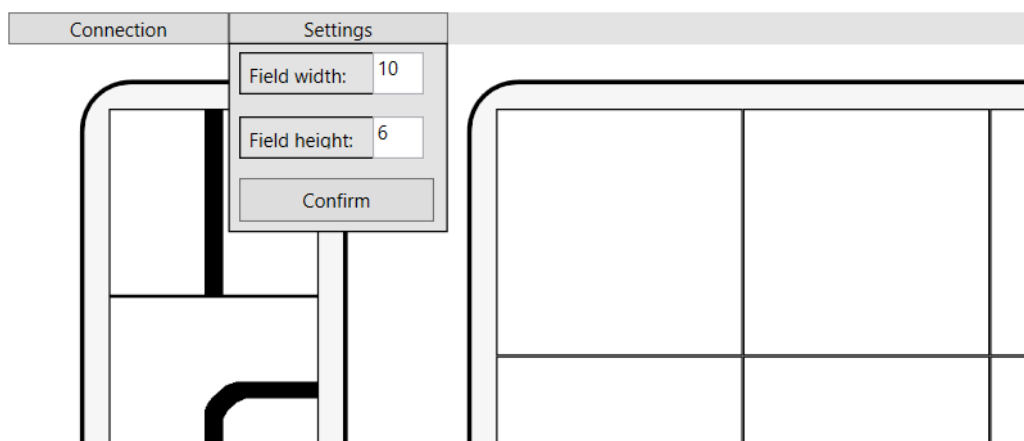
kace - vizuální interpretace dráhy). Na levé straně hlavní plochy jsou na výběr všechny bloky, které se ve stavebnici nacházejí. Uživatel pak může pomocí drag-and-drop systému pokládat jednotlivé bloky na určené místo. Tato funkce umožňuje uživateli naplánovat celou dráhu dopředu, a následně ji sestavit z reálných částí stavebnice.

Základní velikost pole vizuální reprezentace dráhy je  $10 \times 6$  bloků. Tato velikost se dá změnit v záložce „Settings“ (viz obr. 2.6 Záložka „Settings“).

Po přetažení jednotlivých bloků stavebnice do pole je může uživatel přesouvat z jednoho místa na druhé, a otáčet s nimi. Aktivním prvkům (podřízeným modulům) může upravovat vydávanou barevnou kombinaci. Vždy, když tomu tak učiní, aplikace pošle danou změnu do ovládacího modulu.



Obrázek 2.5: Náhled aplikace - vizuální reprezentace dráhy.



Obrázek 2.6: Záložka „Settings“.



## 2.4 Ovládací modul

Ovládací modul je prostředníkem mezi uživatelem, používajícím řídicí aplikaci, a jednotlivými moduly zobrazujícími barevné sekvence. Jeho úkolem je přijímat data z aplikace, tato data zformátovat a poslat na správný podřízený modul ve strukturované podobě. Ovládací modul zároveň slouží jako napájecí zdroj pro celou dráhu.

### 2.4.1 Fyzické provedení

V ovládacím modulu najdeme čtyři hlavní části: baterie s napětím 9 V, regulátor napětí, Arduino Nano, a Bluetooth modul (viz 2.7 Schéma zapojení ovládacího modulu.).

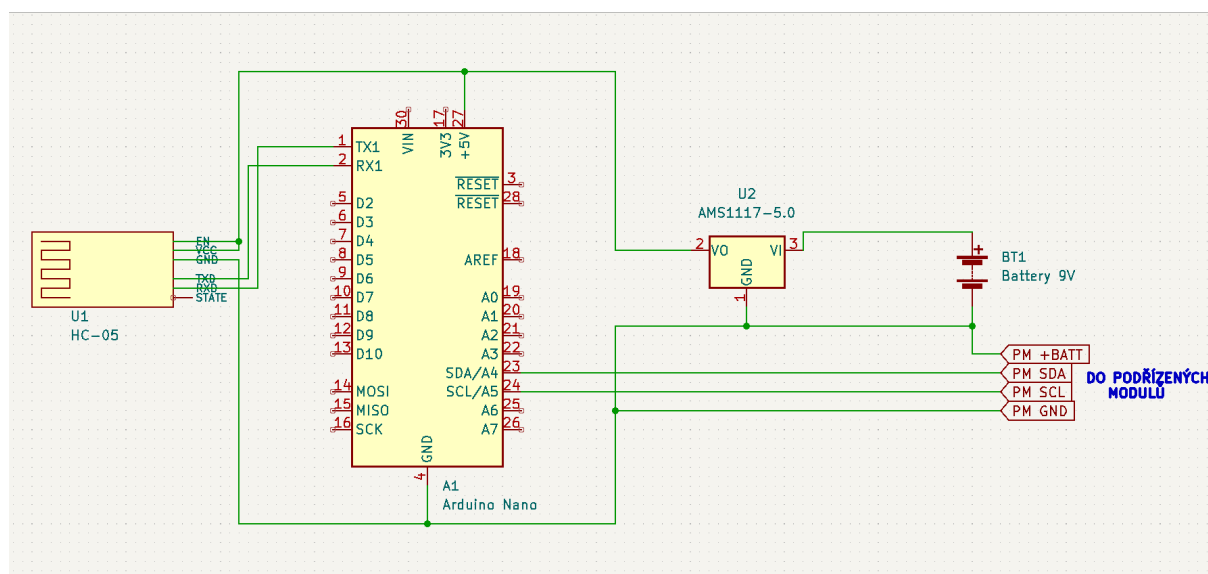
Pro veškeré napájení elektrinou je použita alkalická baterie o napětí 9 V, jakožto běžně dostupná baterie s napětím větším, než 5 V - operačním napětím většiny mikrokontrolérů.

Abychom této elektriny mohli využít, musí projít regulátorem. V modulu je použit regulátor napětí AMS1117 5.0, který ostatní prvky zásobuje stálým napětím 5V. Tento regulátor potřebuje ke správné funkci napětí alespoň 6,5V, modul bude tedy elektrinou zásoben, dokud je baterie nad touto úrovní napětí.

Mozkem ovládacího modulu je platforma Arduino Nano, disponující mikrokontrolérem ATmega328 a vlastním regulátorem napětí na operačních 5V AMS1117 5.0, který je v modulu nevyužitý. Platforma Arduino se stará o veškerá data a jejich zpracování.

Pro komunikaci s řídicí aplikací je v ovládacím modulu přítomen Bluetooth modul HC-05 využívající Bluetooth protokol v2.0 s dosahem cca. 10m.

Veškeré výše zmiňované komponenty jsou uloženy v kompaktní schránce z bílého PET plastu. Schránka má velikost podstavy 55.6 mm × 55.6 mm a výšku 53 mm. Rozměry podstavy byly vybrány pro kompatibilitu modulu s deskou stavebnice LEGO® a to tak, aby zabírala plochu 7 × 7 bodů. Tento rozměr je standardem pro všechny moduly, i pasivní prvky. Výška schránky je definovaná především rozměry baterie, která je umístěna v horní části pro její snadnou výměnu.



Obrázek 2.7: Schéma zapojení ovládacího modulu.

## 2.4.2 Software

Modul přijímá data pomocí výše zmiňovaného Bluetooth modulu ve strukturované podobě, a to následovně: Každá instrukce předaná řídicí aplikací se skládá ze dvou částí - adresy podřízeného modulu, a samotné barevné kombinace. Pro standardizaci dat na vstupu a výstupu, tedy aby byla každá instrukce vždy pětimístná (první místo pro adresu podřízeného modulu, další čtyři pro jednotlivé diody a jejich barvy), je využito tabulky ASCII. Víme-li tedy adresu podřízeného modulu a správné znaky pro rozsvícení diod (viz obr. 2.8 Tabulka znaků pro kontrolu diod.), můžeme poslat podřízenému modulu instrukci o rozsvícení/zhasnutí barev.

	Dioda 1		Dioda 2		Dioda 3		Dioda 4	
	Znak	ASCII	Znak	ASCII	Znak	ASCII	Znak	ASCII
Červená	1	49	5	53	9	57	=	61
Zelená	2	50	6	54	:	58	>	62
Modrá	3	51	7	55	;	59	?	63
Vypnout	4	52	8	56	<	60	@	64

Obrázek 2.8: Tabulka znaků pro kontrolu diod.

Předtím, než tato data pošleme jednotlivým modulům, musí být nejdříve rozdělena na

instrukce pro jednotlivé diody. Instrukce jsou rozděleny v ovládacím modulu, nikoli v modulu podřízeném, abychom zbytečně nezatěžovali slabší mikrokontroléry podřízených modulů. Ty pak přebírají „syrové“ instrukce, a nejsou tak výpočetně zatíženy.

Ovládací modul tedy převezme pomocí Bluetooth data od řídicí aplikace, zjistí z „hlavičky“ těchto dat adresu podřízeného modulu, jednotlivé instrukce rozdělí, dále naváže komunikaci s podřízeným modulem a pošle postupně instrukce o konkrétních diodách a jejich barvách.

## 2.5 Podřízený modul

Podřízený modul je hlavní funkční částí Ozodráhy, jedná se o koncový článek, který pomocí barevných sekvencí komunikuje s Ozobotem. Přijímá data od ovládacího modulu a podle nich upravuje chování čtyř RGB LED diod.

### 2.5.1 Fyzické provedení

Na rozdíl od ovládacího modulu nedisponuje podřízený modul vlastním zdrojem elektřiny. Skládá se ze čtyř hlavních částí: mikrokontroléru, regulátoru napětí, LED diod a schránky, ve které je modul uložen (viz obr. 2.5.1 Schéma zapojení podřízeného modulu.).

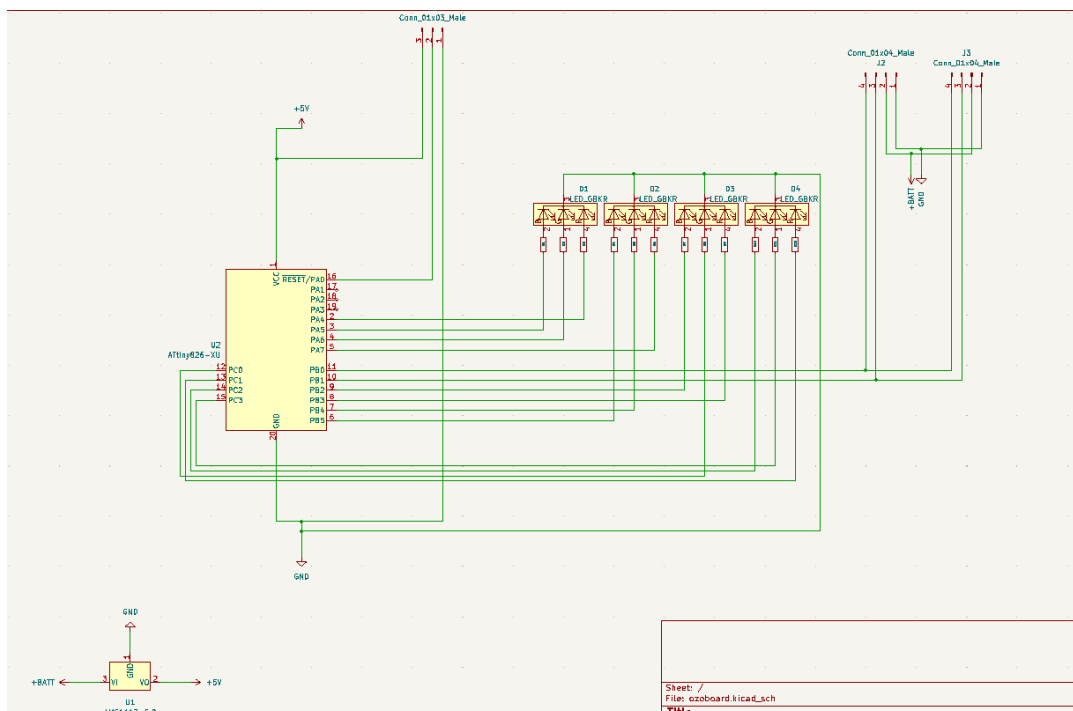
Modul disponuje mikrokontrolérem ATtiny826-XU, který byl vybrán z několika důvodů. Jedná se o mikrokontrolér nejnovější řady rodiny čipů AVR, disponuje 8 kB Flash pamětí a 1 kB SRAM, je tedy více než adekvátní pro plnění funkcí ovládacího modulu. Je navíc ideální díky počtu I/O pinů, mezi nimiž jsou dva přímo určené pro komunikaci[4].

Pro regulátor napětí byl vybrán AMS1117-5.0, který využívá i platforma Arduino. Reguluje poskytovaných 9 V z baterie v ovládacím modulu na 5 V, což je operační napětí výše zmiňovaného mikrokontroléru.

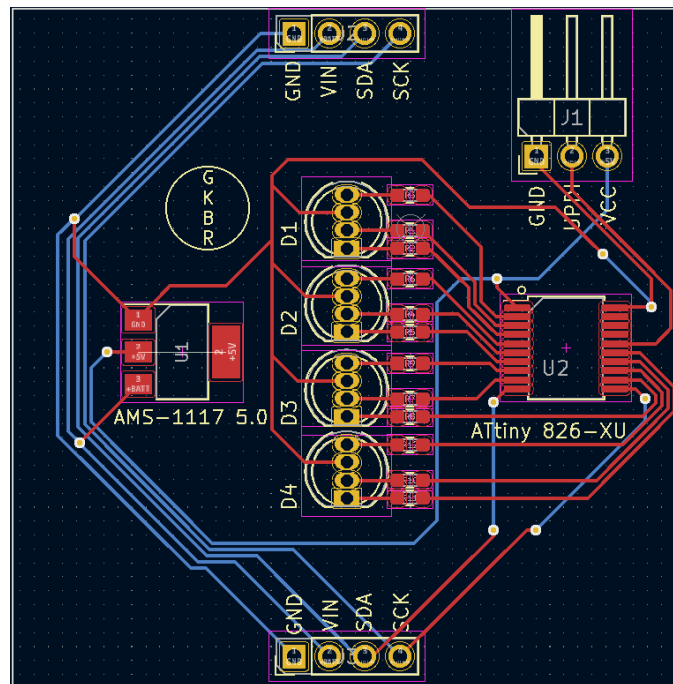
Čtyři RGB LED diody jsou použity pro samotnou komunikaci s Ozobotem. Jelikož je ale světlo vyzařované z těchto diod moc ostré pro barevný senzor, kterým disponuje Ozobot, každá dioda je připojena přes významně větší rezistor, který snižuje dostupné napětí diody, a tím pádem i intenzitu světla.

Tyto tři výše zmiňované části jsou usazeny na dvouvrstvý plošný spoj, který má na dvou okrajích konektory pro komunikaci a napájení (viz obr. 2.5.1 Návrh plošného spoje.).

Stejně jako ovládací modul, i podřízený modul má schránku z bílého PET plastu s tím rozdílem, že schránka pro podřízený modul je navržena tak, aby po její vrchní části mohl jezdit Ozobot. Na ploše vrchní části schránky je tedy proveden výřez, který se v prostřední části prohlubuje, aby skrz tento výřez mohly svítit LED diody. Do tohoto výřezu je zasazen pruh černého PET plastu, který má v prostřední části také řez. Světla z LED diod tedy dosáhnou až na barevný senzor Ozobota, který po tomto černém pruhu jede. Rozměry schránky jsou identické s rozměry schránky ovládacího modulu, tedy  $7 \times 7$  bodů na desce LEGO® ( $55.6 \text{ mm} \times 55.6 \text{ mm}$ ).



Obrázek 2.9: Schéma zapojení podřízeného modulu.



Obrázek 2.10: Návrh plošného spoje.

### 2.5.2 Software

Software nahraný na mikrokontrolér musí být dosti kompaktní kvůli omezené paměti mikrokontroléru. Modul přijímá data od ovládacího modulu. Jelikož jsou data strukturována již v ovládacím modulu a podřízený modul přijímá pouze jednotlivé instrukce, složitost programu není velká. Program pouze přijatá data vyhodnotí podle hodnoty, a rozsvítí/zhasne danou barvu na dané diodě.

## 2.6 Pasivní prvky

Jelikož ne všechny bloky stavebnice potřebují, nebo dokážou upravovat chování Ozobota (aby Ozobot barvy přečetl, musí jet rovně, zatáčky jsou tedy vyloučeny), stavebnice musí být doplněna o pasivní prvky. Jejich jediným úkolem je vést Ozobota, a nahrazovat tak nakreslenou černou čáru na papíře. Pasivními prvky, neboli bloky jsou: zatáčka, rovný úsek, křižovatka „T“ a křižovatka. Tyto bloky mají stejnou šířku a délku jako aktivní

prvky(55.6 mm × 55.6 mm), výšku mají stejnou jako podřízený modul, aby mohl Ozobot přejet z bloků na podřízené moduly bez obtíží, a stavebnice zůstala v jedné rovině.

Bloky jsou sestaveny ze dvou částí, a to spodního rámu a plochy pro Ozobota. Obě části jsou vyrobeny z bílého PET plastu. Jelikož se výška podřízeného modulu několikrát během vývoje změnila, bylo zapotřebí vymyslet řešení, aby se při každém vylepšení a následném snížení výšky nemusely všechny bloky vyrábět znovu. Tímto řešením je spodní rám, který může mít libovolnou výšku. Aby byl použitelný, musí ovšem splňovat tři požadavky, a to: kompatibilita s díly LEGO® na spodní straně, kompatibilita s plochou pro Ozobota na vrchní straně a prostor pro protažení kabelů potřebných ke komunikaci mezi ovládacím modulem a podřízenými moduly.

Plocha pro Ozobota je vrchní část bloku, slouží jako samotná dráha Ozobotu. Stejně jako u podřízeného modulu je v ploše výřez ve tvaru odpovídajícím typu bloku (zatáčka, rovný úsek, křížovatka „T“ a křížovatka), do kterého je zasazen pruh černého plastu, nahrazující samotnou černou čáru.

## 2.7 Komunikace

Jedna z nejdůležitějších částí tohoto projektu je komunikace mezi jednotlivými prvky. Rozlišujeme základní dva směry komunikace, a to mezi aplikací a ovládacím modulem, a mezi ovládacím modulem a podřízenými moduly.

### 2.7.1 Komunikace mezi aplikací a ovládacím modulem

Komunikace mezi řídicí aplikací a ovládacím modulem je zajištěna technologií Bluetooth, který pro zjednodušení a následné snížení nároků na ovládací modul funguje jednostranně, tedy z řídicí aplikace do ovládacího modulu, nikoli opačně. Zařízení s řídicí aplikací vysílá data, ovládací modul je pouze přijímá.

Po zapnutí ovládacího modulu se zapne i Bluetooth modul, a čeká na připojení. V řídicí aplikaci můžeme následně vyhledat tento Bluetooth modul a navázat s ním spojení.

Pokaždé, když se v řídicí aplikaci změni nastavení některého z ovládacích prvků, tato

informace se pošle do ovládacího modulu ve formě stringu.

### 2.7.2 Komunikace mezi ovládacím a podřízenými moduly

Následná komunikace mezi ovládacím modulem a podřízenými moduly je zajištěna přes variaci protokolu I<sup>2</sup>C nazvanou TWI. Jelikož jsou oba mikrokontroléry, jak v ovládacím modulu, tak v podřízeném modulu, z rodiny mikročipů AVR od společnosti Atmel, využívají pro komunikaci stejnou knihovnu „Wire“.

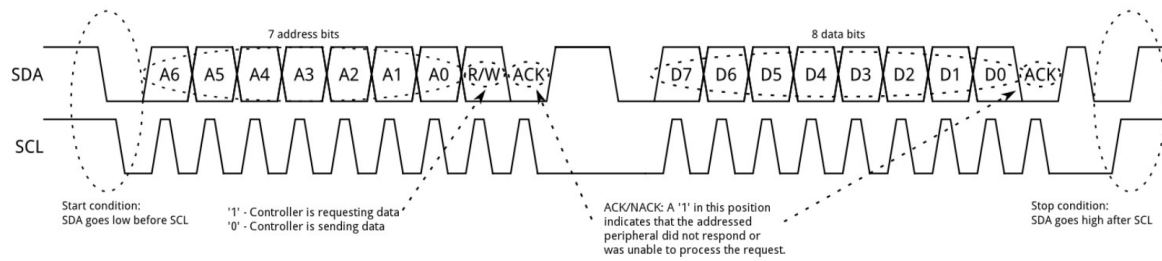
Ovládací modul přijme data z Bluetooth modulu, tyto data zformátuje a následně naváže komunikaci. Každý podřízený modul má pevně danou adresu, na které přijímá instrukce. Adresový rozsah je momentálně od 65 do 90, reprezentovaný v ASCII tabulce znaky „A“ až „Z“. Rozsah je zvolen pro přehlednost, a jelikož je počet podřízených modulů momentálně omezen, není důvod ho rozšiřovat. Možnost tento rozsah rozšířit existuje, např. od 97 do 122, reprezentováno v ASCII tabulce znaky „a“ až „z“.

#### Protokol I<sup>2</sup>C

I<sup>2</sup>C protokol je sériový protokol vyvinutý v r.1982 společností Philips. Hojně se používá k řízení OLED a LED displejů v mobilních telefonech, ovládání hlasitosti chytrých reproduktorů, nebo čtení real-time času. Funguje na principu dvou obousměrných linek, jedné pro hodinový signál(SCL) a jedné pro data(SDA).

Každé vysílání začíná signálem START, kdy je SDA nízká (L) a SCL vysoká (H). Dále následují datové bity, vysílané na lince SDA. Prvních 7 bitů je adresa, osmý je R/W, čili zda chce zařízení psát, nebo číst. Každý byte je ohraničen bitem ACK, který indikuje, zda zařízení odpovídá. ACK bit je vysílán jako H a pokud zařízení odpovídá, přepne v době tohoto bitu SDA na L. Pokud tak neučiní, vysílací zařízení ukončí spojení. Komunikace je ukončena signálem STOP, kdy jsou obě linky vysoké(viz obr. 2.11 I<sup>2</sup>C na časové ose.).

Tento protokol neumožňuje duplexní přenos, vždy může vysílat pouze jedno zařízení, což v případě modulů není problém, jelikož jediný prvek, který vysílá je ovládací modul, podřízený je pouze přijímá na své adrese[5].



Obrázek 2.11: I<sup>2</sup>C na časové ose[7].

### Protokol TWI

TWI (Two-Wire Interface) je protokol prakticky identický k I<sup>2</sup>C, je kompatibilní s mikrokontroléry společnosti Atmel, mezi nimi i ATmega328 používaný v ovládacím modulu, a ATtiny826 používaný v podřízeném modulu[6].



# Kapitola 3

## Závěr

V rámci práce byla navržena a zrealizována modulární stavebnice drah pro roboty Ozobot. V současné době může uživatel sestavovat libovolné dráhy, umisťovat podřízené moduly, kterým pomocí řídicí aplikace může měnit barevné kombinace. Všechny fyzické prvky projektu jsou otestovány a považovány za funkční.



# Literatura

- [1] Prohlídka jazyka C#: přehled [online]. 2022 [cit. 2023-03-23].  
Dostupné z:  
<https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>
- [2] .NET Documentation [online]. [cit. 2023-03-23].  
Dostupné z:  
<https://learn.microsoft.com/en-us/dotnet/>
- [3] What is Windows Presentation Foundation: WPF .NET.[online]. [cit. 2023-03-23].  
Dostupné z:  
<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>
- [4] 59B0F Datasheet: ATtiny424/426/427/824/826/827. [online]. 2021 [cit. 2023-03-23].  
Dostupné z:  
<https://ww1.microchip.com/downloads/en/DeviceDoc/ATtiny424-426-427-824-826-827-DataSheet-DS40002311A.pdf>
- [5] UM10204: ATtiny424/426/427/824/826/827. [online]. 2007 [cit. 2023-03-23].  
Dostupné z:  
[https://web.archive.org/web/20110904124830/http://www.nxp.com/documents/user\\_manual/UM10204.pdf](https://web.archive.org/web/20110904124830/http://www.nxp.com/documents/user_manual/UM10204.pdf)
- [6] What is TWI?. [online]. 2018 [cit. 2023-03-23].  
Dostupné z:  
<https://ww1.microchip.com/downloads/en/DeviceDoc/90003181A.pdf>

[7] I2C: Sparkfun. Sparkfun [online]. Niwot, Colorado [cit. 2023-03-22].

Dostupné z:

<https://learn.sparkfun.com/tutorials/i2c/all>

[8] FOOT, Peter. 32feet.NET: InTheHand. In The Hand Ltd [online]. [cit. 2023-03-22].

Dostupné z:

<https://inthehand.com/components/32feet/>

# Seznam obrázků

1.1	Tabulka sekvencí. . . . .	10
2.1	Schéma architektury. . . . .	12
2.2	Schéma sestavené dráhy. . . . .	13
2.3	Záložka „Connection“. . . . .	15
2.4	Přijímání dat od uživatele. . . . .	15
2.5	Náhled aplikace - vizuální reprezentace dráhy. . . . .	16
2.6	Záložka „Settings“. . . . .	16
2.7	Schéma zapojení ovládacího modulu. . . . .	18
2.8	Tabulka znaků pro kontrolu diod. . . . .	18
2.9	Schéma zapojení podřízeného modulu. . . . .	20
2.10	Návrh plošného spoje. . . . .	21
2.11	I <sup>2</sup> C na časové ose[7]. . . . .	24