



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

Fakulta elektrotechnická

Katedra kybernetiky

Pomůcka pro rehabilitaci (cvičení) pohyblivosti okohybných svalů

Aid for Rehabilitation (exercise) of Oculomotor Muscle Mobility

## **BAKALÁŘSKÁ PRÁCE**

**Jméno a příjmení autora**

Martin Vrňata

Studijní program: Kybernetika a robotika

Vedoucí bakalářské práce: Ing. Petr Novák Ph.D.

**Praha, Květen 2023**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vrňata** Jméno: **Martin** Osobní číslo: **491893**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra kybernetiky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Pomůcka pro rehabilitaci (cvičení) pohyblivosti okohybných svalů**

Název bakalářské práce anglicky:

**Aid for Rehabilitation (exercise) of Oculomotor Muscle Mobility**

Pokyny pro vypracování:

Po mechanických úrazech hlavy, nemocích, ale i některých psychických stavech dochází k poruše / omezení činnosti okohybných svalů a tím i k nesprávnému (často dvojitému) vidění. Základem nápravy tohoto nežádoucího stavu je důkladné cvičení okohybných svalů. Tato cvičení jsou většinou formou sledování pohybujícího se stimulačního bodu.

1. Prostudujte již existující zařízení sloužící ke cvičení okohybných svalů. Zhodnoťte jejich možnosti v poskytování různých typů trajektorií stimulačního bodu (čára vodorovně / šikmo, kruh, elipsa, ...). Rovněž zhodnoťte i poskytnutý rozsah pohybu stimulačního bodu.

2. Navrhněte a vytvořte prototyp zařízení umožňující velký rozsah pohybu stimulačního bodu formou jeho promítání například na stěnu. Navrhněte jednoduchý způsob zadávání trajektorie stimulačního bodu obsluhou, poté výpočet pohybu po trajektorii, její interní reprezentaci a případně uložení pro opakované použití.

3. Navrhněte možnost jednoduché obsluhy přístroje. Tedy nastavení trajektorie a jejich parametrů (rozsah / rychlost pohybu stimulu). Myslete na bezpečnost obsluhy v případě použití laserového ukazovátko jako promítaného stimulu.

4. Zhodnoťte výhody / nevýhody tohoto vlastního řešení oproti již existujícím, zejména z pohledu možností a snadnosti ovládání.

Seznam doporučené literatury:

[1] Noviello Carmine, Mastering STM32, LeanPub, 2017 ([www.carminenoviello.com](http://www.carminenoviello.com))

[2] Majid Pakdel, Advanced Programming with STM32 Microcontrollers, Elektor, 2020, 978-3-89576-410-3

[3] Price Mark, C# 8.0 and .NET Core 3.0 - Modern Cross-Platform Development, Packt, 2019

[4] WWW stránky / projekty zabývající se tímto problémem / tématikou

Další potřebné materiály poskytnete vedoucí práce.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Petr Novák, Ph.D. oddělení kognitivních systémů a neurovědy CIIRC**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **05.01.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Petr Novák, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

### **Prohlášení autora práce**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne .....

.....

Martin Vrňata

## Poděkování

Rád bych poděkoval panu Ing. Petrovi Novákovi Ph.D. za odborné vedení práce, přátelský přístup a cenné rady, které mi velice pomohly při psaní práce a vývoji zařízení. Dále také děkuji spolužákovi Václavu Rissovi za technické rady a psychickou podporu při studiu a psaní této práce.

## Abstrakt

Tato práce se zabývá tvorbou prototypu zařízení pro rehabilitaci okohybných svalů. Vytvářené zařízení se skládá ze dvou hlavních částí: promítací a ovládací. Promítací část vytváří na stěně pohyblivý stimulační bod, který pacient sleduje, a tím rehabilituje / procvičuje okohybné svaly. Ovládací část slouží k řízení promítací části, tedy výběru a nastavení trajektorií stimulačního bodu na stěně. Uživatel si může vybrat ze 4 typů trajektorií, „Přímky“, „Křivky“, „Vlastní“ a „OnLine“.

**Klíčová slova:** Bakalářská práce, .NET MAUI, C, C#, STM32CubeIde, MS Visual Studio, modelářská serva, NUCLEO, laser, Bluetooth

This work deals with the creation of a device prototype for the rehabilitation of oculomotor muscles. The created device consists of two main parts, projection and control. The projection part creates a moving stimulation point on the wall, which the patient follows with their sight, thereby rehabilitating / exercising the oculomotor muscles. The control part is used to control the projection part, select and set the trajectories of the stimulation point on the wall. The user can choose from 4 types of trajectories, "Lines", "Curves", "Custom" and "OnLine".

**Key words:** Bachelor thesis, .NET MAUI, C, C#, STM32CubeIde, MS Visual Studio, modeling servo. NUCLEO, laser, Bluetooth

## Obsah

1	Úvod .....	2
2	Současný stav.....	4
3	Cíle práce .....	6
4	Názvosloví.....	7
5	Návrh řešení .....	8
5.1	Celková činnost.....	8
5.2	Laser s optikou – světelný zdroj.....	8
5.3	Pohybový mechanismus .....	9
5.4	Procesorová deska.....	10
5.5	Ovládací zařízení .....	11
5.6	Komunikace .....	12
5.7	Ovládací aplikace .....	14
6	Implementace řešení.....	15
6.1	Promítací zařízení .....	15
6.1.1	Ovládání pohonů .....	16
6.1.2	Ovládání laseru .....	16
6.1.3	Signalizační LED .....	17
6.1.4	Trajektorie.....	17
6.2	Aplikace .....	20
6.3	Tvorba a zpracování trajektorie .....	22
6.3.1	Lines – čárový průběh.....	22
6.3.2	Curves – parametrické křivky .....	23
6.3.3	Custom – vlastní definované křivky .....	24
6.3.4	OnLine – přímé řízení stimulačního bodu.....	25
7	Návod k použití .....	27
8	Závěr .....	34
9	Zdroje.....	35

### Seznam příloh

.apk soubor mobilní aplikace – Aktuální verze aplikace a všechny ostatní soubory jsou u vedoucího práce.

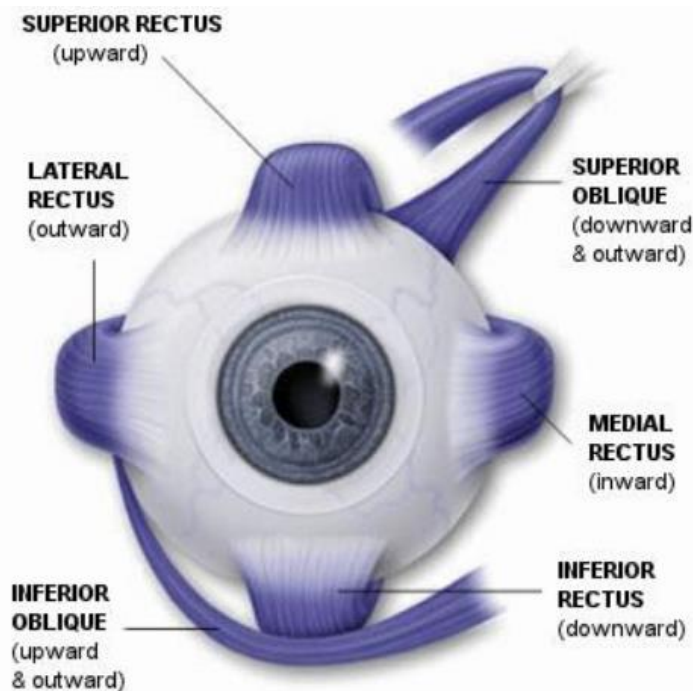
# 1 Úvod

V současné době u lidí dochází stále více k různým typům úrazů. Některé z těchto úrazů mohou poškodit nebo omezit činnost okohybných svalů a tím způsobit nesprávné vidění, což může být pro člověka velmi nepříjemné. Nejčastější příčiny způsobující poškození činnosti okohybných svalů, jsou na příklad:

- **Úraz hlavy** – Člověk se udeří do hlavy a poškodí si některé okohybné svaly.
- **Stres** – Stres může způsobit nerovnoměrné zatěžování okohybných svalů a dlouhodobě může dojít až k jejich skutečnému poškození.
- **Časově dlouhé dívání do obrazovky** – Dívá-li člověk dlouho na obrazovku, ať už televizní či počítačovou, dochází k nerovnoměrnému zatěžování okohybných svalů a může rovněž dojít k jejich poškození.

Takovéto poškození okohybných svalů může mít za následek v nejhorším případě i dvojité vidění. U lehčích případů se může jednat o neostré vidění nebo o neestetické ujždění jednoho oka stranou. Poškození okohybných svalů může rovněž způsobovat silné migrény či bolení očí samotných, případně bolesti hlavy.

V lehčích případech, se poškození okohybných svalů napravuje pouze rehabilitací / procvičováním, avšak těžší poškození se většinou řeší operací. Operace je oproti rehabilitaci invazivní, právě proto se preferuje rehabilitace, tedy cvičení.



Obrázek 1 – Šestice svalů pro pohyb oka v různých pohledových směrech

Na obrázku 1 jsou vidět okohybné svaly určené k pohybu oka do všech potřebných směrů. Je zde vidět 6 okohybných svalů, které zajišťují pohyby oka nahoru, dolů, doleva, doprava a případně i částečné natočení.

Základem rehabilitačního typu nápravy poškození činnosti okohybných svalů je právě důkladné a pravidelné procvičování zejména těch poškozených svalů. Podle typu poškození konkrétních svalů musí pacienti vykonávat odpovídající pohyby očí. Někteří pacienti potřebují cvičit očima v podstatě pouze



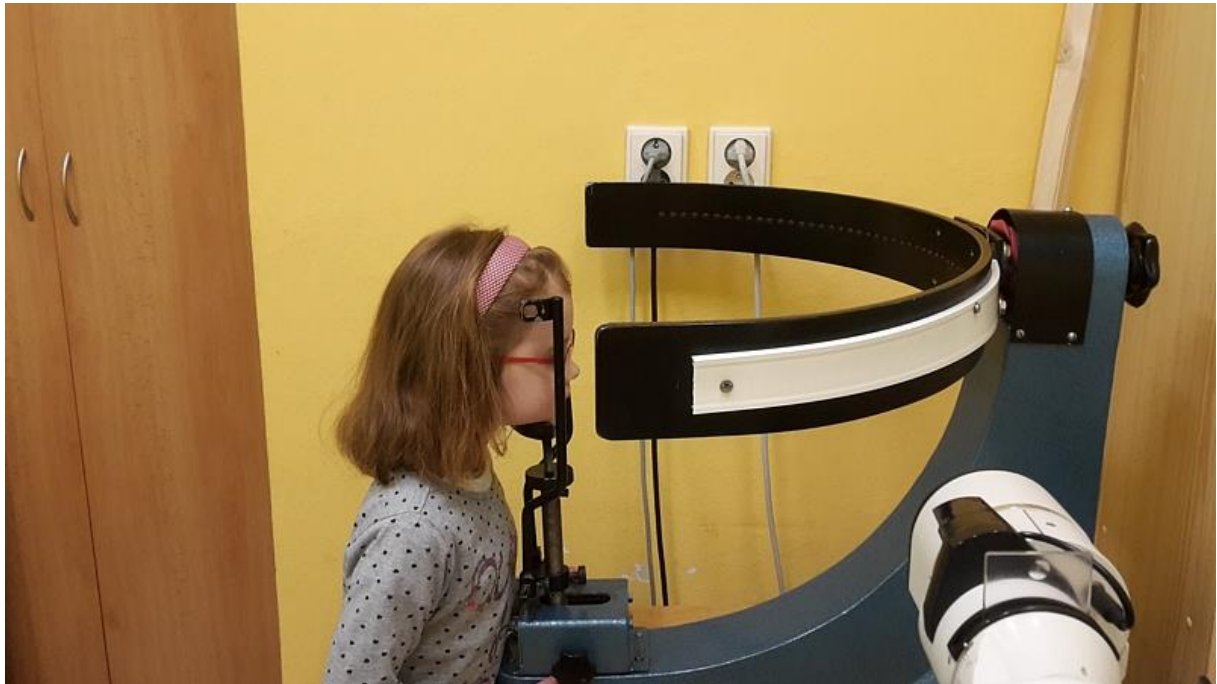
zleva - doprava, někteří naopak pouze nahoru - dolů. Jiní zase procvičují všechny okohybné svaly současně, a tudíž je pro ně vhodné pohledem kopírovat elipsu.

V naprosté většině případů tedy pacient sleduje nějaký stimulační bod pohybující se tak, aby se procvičily potřebné poškozené okohybné svaly.

## 2 Současný stav

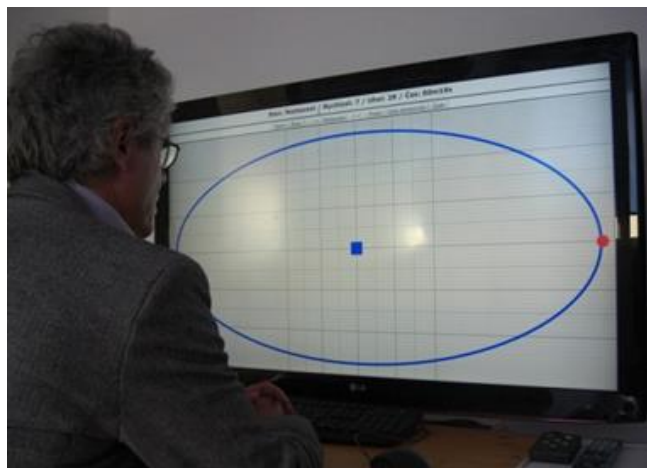
V současnosti se ke zmíněné rehabilitaci používají zejména celkem drahé a velké pomůcky, a tudíž musí pacient pravidelně / často docházet do ordinace očního lékaře.

- **Pás LEDek** – V řadě poskládané, postupně se po jedné rozsvěčující LEDky a takto vytvářející pohybující se stimulační bod. Tato rehabilitační pomůcka je poněkud nákladná a zabírá nemálo místa. Navíc je rozsah pohybu očí omezen vždy pouze do jedné osy a chce-li pacient změnit natočení této přímky, musí si fyzicky přístroj pře-nastavit. Vzhledem k ceně a rozměrům je tento přístroj pro domácí cvičení v podstatě nedostupný / nepoužitelný. Bohužel často i pro lékařské ordinace.



Obrázek 2 – Dítě cvičící okohybné svaly pomocí přístroje obsahující (pouze) pás LEDek

- **Sledování pohyblivého stimulačního bodu na obrazovce (například TV)** – Tato forma rehabilitace řeší cenu a částečně i velikost předešlé verze, jelikož skoro každý má doma počítač s obrazovkou (případně TV). Problém je ovšem stále omezený rozsah cvičení a potřeba být blízko k obrazovce, což může oči naopak i zatěžovat.



Obrázek 3 – Cvičení okohybných svalů pomocí TV (vlevo), Nastavení trajektorie pro cvičení na TV (vpravo) [4]

Pacienti dochází do ordinace většinou jednou týdně na rehabilitaci trvající až hodinu (ne pouze cvičení očních svalů). Tento stav není pro pacienty ideální. Ve skutečně ideálním případě by měli pacienti rehabilitovat poškozené okohybné svaly každý den, a to i několikrát v rámci dne (například ráno i večer).

Shrnutí současného stavu:

- **Nutnost docházet do ordinace** – Pacient musí pravidelně docházet rehabilitovat do ordinace. To pro některé lidi není vhodné, např. pro lidi s jiným tělesným postižením (pacient hůře chodí) nebo bydlí celkem daleko od lékaře.
- **Pacient nemůže rehabilitovat doma** – Kdyby pacient mohl rehabilitovat doma, ušetří se čas doktora, ten se nemusí věnovat (pouze) rehabilitujícímu pacientovi, ale může se věnovat jiným závažnějším případům. Zároveň v pohodlí domova může rehabilitovat více pacientů současně.
- **Rozměry** – Současné pomůcky pro rehabilitaci očních svalů jsou poněkud velké a tedy zabírají v ordinaci spoustu místa.
- **Cena** – Rehabilitační pomůcky okohybných svalů jsou velice drahé, a tudíž i pro některé ordinace těžko dostupné.

### 3 Cíle práce

Cíle práce lze shrnout do následujících bloků:

- **Zhodnotit současný stav** – Výhody a nevýhody současných řešení (popsáno v úvodu).
- **Navrhnout okohybný treňažer a jeho činnost** – Navrhnout treňažer okohybných svalů tak, aby byl vhodný k rehabilitaci pro co nejvíce uživatelů.
- **Realizovat prototyp treňažeru (promítacího zařízení)** – Naprogramovat procesor a rovněž k němu připojit i další komponenty treňažeru.
- **Vytvořit ovládaní treňažeru (ovládací aplikaci)** – Vytvořit ovládací aplikaci a navrhnout způsob komunikace mezi ovládacím a promítacím zařízením.
- **Zhodnocení** navrženého řešení – Zhodnotit přínosy a rovněž (možné) nevýhody zde navrženého řešení / zařízení.

Cílem práce je navrhnout a vytvořit prototyp rehabilitačního zařízení pro pacienty s poškozenými okohybnými svaly. Takové zařízení by mělo být cenově dostupné, aby si jich doktor mohl do své ordinace pořídit i několik a případně je rovněž půjčovat pacientům domů. Dále takováto rehabilitační pomůcka musí být celkem malá, aby nezabírala v ordinaci, či u pacienta doma, příliš místa. Má-li zařízení používat pacient doma, je rovněž klíčová jeho jednoduchá ovladatelnost. Rehab. pomůcka by tedy měla být jednoduše nastavitelná a také snadno obsluhovatelná.

Vytvářené zařízení by mělo být v podstatě „krabička“ promítající pohyblivý stimulační bod pomocí nějakého světelného zdroje (např. laseru) na stěnu či plátno. Toto zajistí dostatečně velký rozsah pohybu stimulačního bodu. Pacient pak bude stimulační bod sledovat z vhodné vzdálenosti a bude tedy správně procvičovat své okohybné svaly.

Pacient by měl být schopen (na dálku) ovládat trajektorii pohybu stimulačního bodu (tvar/rychlost), vybrat si trajektorii z několika přednastavených a rovněž mít možnost si vytvořit zcela vlastní trajektorii. Přednastavené trajektorie musí být dostatečně rozmanité, aby si pacient mohl vybrat trajektorii vyhovující stavu poškození okohybných svalů.

Zařízení a jeho obsluha musí být navrženy tak, aby se minimalizovala pravděpodobnost posvícení světelným zdrojem (např. tedy laserem) pacientovi do očí, což by mohlo být pro pacienta nepříjemné.

Rehabilitační pomůcka by rovněž měla sledovat dobu rehabilitace, aby pacient věděl, jak dlouho již cvičí. Příliš dlouhé / nadměrné cvičení nemusí být vždy přínosem.

Práce se naopak nevěnuje následujícím částem / oblastem:

- Jedná se pouze o prototyp zařízení, a tudíž výstupem práce není kompletní dokumentace.
- Komunikace mezi řídicím zařízením a promítacím zařízením není nikterak zabezpečena.
- Při tvorbě zařízení je kladen důraz na jeho zejména technickou činnost nikoli medicínskou způsobilost a oprávněnost.

## 4 Názvosloví

Pro přehlednost dalšího textu byly stanoveny následující pojmy:

- **Trenažér okohybných svalů** – Pomůcka (dnes většinou mechanického typu) pro cvičení okohybných svalů.
- **Promítací zařízení** – Stacionární zařízení vytvářející pohyb promítaného stimulačního bodu na cílovou plochu (například zeď).
- **Ovládací zařízení** – Osobní zařízení pro uživatele / obsluhu sloužící k nastavení trajektorie a celkového ovládní promítacího zařízení.
- **Pohybový mechanismus** – Soustava dvou aktuátorů v ose X a Y sloužící pro skutečný pohyb (například laserovým ukazovátkem).
- **Procesorová (řídící) deska** – Centrální deska obsahující procesor a podpůrné obvody a zajišťující / řídící činnost celého promítacího zařízení.
- **Logické souřadnice** – Souřadnice bodů používané v ovládacím zařízení a pro přenos bodů / trajektorie z ovládacího zařízení do promítacího zařízení. Tyto souřadnice jsou v intervalu [-100; +100].
- **Fyzické souřadnice** – Souřadnice vytvářené na cílové ploše dané fyzickými možnostmi pohybového mechanismu (např. výchylkou serva pomocí PWM hodnoty).

## 5 Návrh řešení

### 5.1 Celková činnost

Trenažér je celkem malé zařízení umístěné za pacienta (ideálně ve výši hlavy) proti dostatečně prostorné stěně před očima / pohledem pacienta. Předpokládaná vzdálenost trenažéru od promítací stěny jsou 1.5 až 3 metry. V zařízení se nachází dva rotační aktuátory hýbající se světelným zdrojem v rotačních osách X a Y. Takto pohyblivý světelný zdroj poté promítá stimulační bod na cílovou stěnu, tedy před oči pacienta. Stimulační bod se pohybuje po zadané trajektorii (možno oběma směry). Zařízení dále obsahuje procesorový modul, signalizační LED a ovládání světelného zdroje. K procesorové desce je rovněž připojen komunikační modul pro komunikaci s ovládacím zařízením. Napájení celého zařízení bude zajišťovat vhodný síťový adaptér.

Z pohledu uživatele bude (promítací) zařízení vytvářet pohyb stimulačního bodu v těchto základních režimech:

- **Průběh složený z přímk** (Lines) – Průběh se skládá ze zadaných bodů a ty jsou následně propojeny přímkami.
- **Složitější křivky** (Curves) – Sofistikovanější křivky (elipsa, ležatá osmička, ...). Uživatel může nastavit střed křivky a parametry, které křivku definují (velikost/tvar).
- **Zcela vlastní** (Custom) – Uživatel si nakreslí trajektorii podle vlastní potřeby.
- **OnLine režim** – Uživatel z ovládacího zařízení přímo hýbe s bodem promítaným na stěnu.

Tříbarevná signalizační LED oznamuje uživateli v jakém stavu se zařízení aktuálně nachází. Zařízení se může nacházet ve 3 stavech signalizovaných diodou třemi barvami, červenou, zelenou a modrou (popsáno v kapitole 7 Návod k použití).

Uživatel ovládá trenažér pomocí ovládacího zařízení (buď přímo na trenažéru, připojeným kabelem k trenažéru, či bezdrátově). Pacient si pomocí ovládacího zařízení může vybrat buď přednastavenou trajektorii, nebo si vytvoří zcela vlastní. Následně je pak navolená trajektorie pomocí vhodně vytvořené komunikace odeslána do promítacího zařízení, kde je trajektorie zpracována a připravena k vykonání, tedy pohybu stimulačního bodu.

Pacient si rovněž může nastavit dobu trvání celé rehabilitace. Poté může být zařízení spuštěno, zastaveno, nebo se dá kdykoli otočit směr pohybu po trajektorii, což je velmi vhodné pro časově delší cvičení.

Pacient následně sleduje pohybující se stimulační bod na stěně, a tím tedy procvičuje okohybné svaly, dokud nevyprší doba nastavené rehabilitace, nebo dokud se pacient nerozhodne rehabilitaci ukončit dříve.

V průběhu rehabilitace promítací zařízení očekává každých 15s příjem udržovacího packetu nazvaného „Check“. Nepřijde-li tento paket za uvedený čas, zařízení z bezpečnostních důvodů nejen vypne serva, ale zejména zhasne laser.

### 5.2 Laser s optikou – světelný zdroj

Stimulační bod je promítán na zeď pomocí (pohybujícího se) laseru. Lasery mohou mít spoustu barev a různé (světelné) výkony. Je potřeba vybrat takový laser, jehož promítaný bod bude vidět na nejčastějších barvách stěn i v běžných denních světelných podmínkách. Výkon laseru tedy musí být dostatečně velký, ale zároveň nesmí ohrožovat oči pacienta (při náhodném a krátkém svitu do oka). Vybral jsem červený laser, jelikož je nejdostupnější (a nejlevnější) a zároveň je tato barva pro oko vcelku příjemná, a tedy dobře viditelná. Použitý laser má následující parametry:

- Napájecí napětí: 5V
- Odběr proudu: 20 mA
- Výstupní výkon: < 5mW
- Třída laseru: IIIA
- Vlnová délka světla: 650nm



Obrázek 4 - Použitý laser s optikou (laskakit.cz)

### 5.3 Pohybový mechanismus

Aby se stimulační bod promítaný laserem pohyboval, je třeba s laserem vhodně hýbat / vychylovat. Laser je tedy připevněn na soustavu aktuátorů umožňující jeho pohyb současně ve dvou rotačních osách, vodorovné a svislé. Pro moje použití připadají v úvahu dva typy pohonů. Krokové motory a (modelářská) serva.

Krokové motory nejsou nikterak levné, mají však dlouhou mechanickou výdrž a jsou tiché při malých rychlostech. Jejich hlavní nevýhoda je malý počet poloh / kroků na otáčku, do kterých se dá motor vystavit. To omezuje rozlišení při promítání bodů / pozic na stěně. Lze použít krokové motory mající větší počet poloh / kroků na otáčku, ale jejich cena je již značně vyšší. Případně využít vhodnou převodovku. Poslední možností je použít tzv. mikro-krokování pro vytvoření dílčích mezi-kroků k již existujícím. Dále při použití krokových motorů může docházet k cukání stimulačního bodu při jeho pohybu po stěně (velmi rychlé, až skokové, vystavení do nové polohy).

Alternativou jsou (modelářská) serva jejichž životnost však nemusí být tak dlouhá jako u krokových motorů (ozubené převody, interní potenciometry). Při vysokých rychlostech mohou být až hlučná. Tato serva se řídí pomocí PWM signálu, což je celkem jednoduché. Dají se tedy (v podstatě teoreticky) nastavit do mnohem více poloh / úhlů než krokové motory (záleží na typu a ceně serv). Hluk serv by měl být dostatečně eliminován krabičkou, ve které se budou nacházet.

Pro vytvářený prototyp jsem tedy vybral (modelářská) serva kvůli jejich jemnému pohybovému rozlišení. Budou-li moc hlučná, nebo jejich životnost nebude dostačující, lze je v budoucnu vyměnit za mnohem kvalitnější typy, a tudíž i dražší, nebo případně za krokové motory s vyšším pohybovým / krokovým rozlišením.

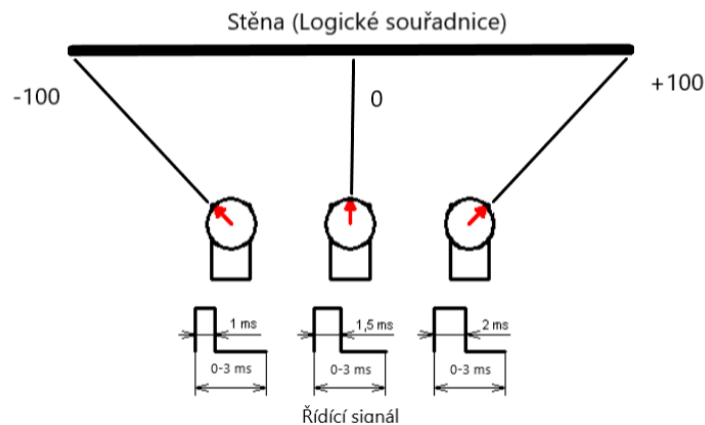
Vybraný typ (modelářského) serva nese označení HS-5485HB. Pro moje prototypové zařízení jsou tato serva poněkud velká a v budoucnu budou nejspíš nahrazeny menším modelem (pro prototyp však velmi vhodná). Parametry lze stručně shrnout následovně:

- Napájecí napětí: 4.8 V
- Základní frekvence PWM: 333 Hz
- Rozsah střídý pro PWM: 1-2 ms (pro rozsah otočení 0° – 90°)
- Tah: 5.2 kg/cm
- Rozměry 39.8 x 19.8 x 38 mm
- Hmotnost: 45 g



Obrázek 5 - Modelářské servo HS-5485HB

Z obrázku 5 je zřejmé natočení serva při promítání stimulačního bodu na stěnu v závislosti na Logických souřadnicích. Logická pozice -100 odpovídá natočení nejvíce vlevo (-45°) a tedy maximální dosažené fyzické pozici promítaného bodu do levé strany. Naopak logická pozice +100 odpovídá natočení nejvíce vpravo (+45°) a tedy maximální dosažené fyzické pozici promítaného bodu do pravé strany. Maximální fyzický rozsah / rozměr promítaného obrazu závisí samozřejmě na vzdálenosti promítacího zařízení od cílové stěny.



Obrázek 6 - Vztah mezi střídou řídicího PWM signálu serv a Logickými souřadnicemi na cílové stěně

#### 5.4 Procesorová deska

Pohyb stimulačního bodu po stěně je řízen procesorovou deskou vykonávající patřičný program. Na této desce se počítá pohyb po zadané trajektorii na fyzické stěně a je tedy důležité, aby měla vhodně velkou RAM paměť. Dále je důležité, aby použitý procesor měl rovněž dostatečný výpočetní výkon, aby byla dosažena uspokojivá jemnost pohybu stimulačního bodu. Základní požadavky na procesorovou desku, v podstatě osazený procesor, jsou tedy následující:

- Dostatečná RAM pro před-počítání a uložení trajektorie
- Paměť FLASH pro řídicí program
- 2x výstup PWM pro řízení serv
- Výstup pro zapínání/vypínání laseru
- Výstup pro signalizační třibarevnou LEDku
- Podpora komunikace (výstup pro komunikační modul I2C/SPI/UART...) s ovládacím zařízením

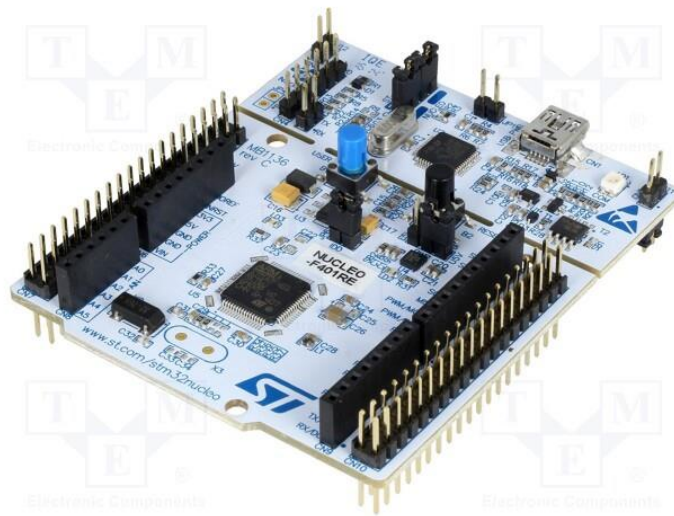
Pro vytváření prototyp zařízení byla vybrána (vývojová) procesorová deska Nucleo-F401RE s procesorem STM32F401RET6 s jádrem ARM Cortex M4 [9].

Tento procesor má následující vlastnosti:

- **Taktovaný na 84 MHz**, což by mělo být dostatečné pro mé použití.
- **FLASH paměť o velikosti 512 kB** pro uložení řídicího programu.
- **96 kB SRAM** pro uložení před-počítané trajektorie.
- Má více jak **2 výstupy pro PWM signál**. A spoustu výstupních pinů pro LED či tlačítka.
- **Sběrnice SPI/I2C/...** pro připojení různých externích periférií / modulů.
- Napájení 3.3V, většina pinů je 5V tolerantních.



Tyto parametry jsou více než dostačující pro mojí aplikaci a později se může použít i levnější procesor. Vývojová deska obsahuje rovněž přímo i programátor a proto je velmi vhodná pro tvorbu / vývoj prototypu zařízení.



Obrázek 7 - Vývojová deska Nucleo-F401RE použitá pro prototyp zařízení

## 5.5 Ovládací zařízení

Okohybný trenažér, jako elektronické/mechanické zařízení, je samozřejmě potřeba nějak vhodně ovládat. Dále jsou uvedeny některé uvažované způsoby ovládání a jejich výhody / nevýhody:

### 1. Pouze ovládací tlačítka přímo na zařízení trenažéru:

- (+) Jednoduché ovládání, jednoduchá tvorba programového vybavení.
- (-) Není možnost (snadno) si vytvořit vlastní trajektorie. Uživatel si musí vybrat z trajektorií přednastavených.
- (-) Lze ovládat pouze základní činnosti.
- (-) Uživatel se při ovládání musí otočit čelem na trenažér a tím se zvyšuje riziko, kdy by mohl laser posvítit uživateli do očí.
- (-) Uživatel nemůže ovládat trenažér na dálku (například ze židle).

### 2. Dotykový (grafický) display přímo na zařízení trenažéru:

- (+) Uživatel si může nastavit / vytvořit vlastní trajektorie.
- (+) Komplexní ovládání trenažéru – možnost tvorby grafického rozhraní.
- (-) Uživatel se při ovládání musí otočit čelem na trenažér a tím se zvyšuje riziko, kdy by mohl laser posvítit uživateli do očí.
- (-) Zvýší se pořizovací cena trenažéru (není však kritické).
- (-) Trenažér se nedá ovládat na dálku (může být bráno jako omezení).

### 3. Připojení zařízení trenažéru například na osobní / stolní počítač:

- (+) Výpočetní kapacita počítače a bohaté grafické rozhraní.
- (+) Komplexní ovládání trenažéru – snadnost vývoje ovládací aplikace.
- (-) Ne každý počítač má wifi nebo Bluetooth. Spojení kabelem (USB) není vhodné z hlediska bezpečnosti pohybu kolem zařízení.
- (-) Ne každý má počítač, a kdyby měl být počítač součástí trenažéru, tak se zvedne nepřiměřeně celková pořizovací cena.

### 4. Chytrý mobilní telefon / tablet:

- (+) Skoro každý má chytrý telefon (případně i tablet).

- b. (+) Komplexní ovládání trenažéru – možnost vývoje plně grafické aplikace.
- c. (+) Výpočetní kapacita chytrého telefonu a bohaté grafické rozhraní.
- d. (+) Každý chytrý telefon má Bluetooth a wifi.
- e. (-) Chytrý telefon je napájen z baterie, což může limitovat dlouhou dobu rehabilitace například v ordinaci (lze řešit dobíjením).

Pro ovládání prototypu trenažéru jsem zvolil chytrý mobilní telefon / tablet. Tuto možnost jsem vybral zejména z důvodu mnoha dostupných typů komunikací a možnosti vytvořit ovládací aplikaci s bohatým uživatelským grafickým rozhraním. Také tato volba nikterak nezvyšuje pořizovací cenu trenažéru (mobilní telefon nebude součástí trenažéru).

## 5.6 Komunikace

Ovládací zařízení musí komunikovat s promítacím zařízením za účelem jeho ovládání / řízení. Způsobů komunikace, připadajících v úvahu, je několik. Např.

### 1. USB

- a. (+) Spolehlivost – Při komunikaci přes USB je malá pravděpodobnost ztráty nebo poškození dat.
- b. (-) Uživatel musí být připojený k zařízením kabelem a tudíž musí být relativně blízko promítacího zařízení.
- c. (-) Nebezpečí „zamatání“ se do kabelového spojení.

### 2. Wifi

- a. (+) Vysoký dosah – Promítací zařízení může komunikovat s ovládacím zařízením na velkou vzdálenost. Toto může být současně i nevýhoda, jelikož uživatel může odejít od trenažéru poměrně daleko a trenažér zůstane zapnutý (zejména jeho laser).
- b. (+) Každý chytrý telefon / tablet má wifi.
- c. (-) Potřeba Wifi sítě – Aby spolu mohly ovládací a promítací zařízení komunikovat, je potřeba zajistit Wifi point, ke kterému se obě zařízení připojí. Případně musí Wifi síť vytvořit přímo zařízení.
- d. (-) Vysoká spotřeba energie.
- e. (-) Komunikace přes Wifi není u procesorů s architekturou ARM běžná (nutnost sofistikovanějšího externího modulu).

### 3. Bluetooth

- a. (+) Dosah do 10m (u Bluetooth Low Energy) – Komunikace přes Bluetooth (LE) nemá takový dosah jako komunikace přes wifi – To je zde výhodou, jelikož vzdálí-li se uživatel s ovládacím zařízením příliš daleko, komunikace se přerušuje a trenažér se zastaví (laser se vypne). Zároveň komunikace přes Bluetooth umožňuje snadné vzájemné přímé spojení dvou zařízení.
- b. (+) Každý chytrý telefon / tablet má Bluetooth
- c. (-) Nespolehlivost – Možná ztráta informací při komunikaci (nutno zajistit pomocí SW).
- d. (-) Nutnost párování – U Bluetooth Low Energy se zařízení párovat nemusí (velká výhoda).
- e. (-) Komunikace přes Bluetooth není u procesorů s architekturou ARM zcela běžná (lze doplnit velmi jednoduchý externí modul).

Pro prototyp jsem vybral komunikaci pomocí přídavné desky s Bluetooth Low Energy s označením X-Nucleo - IDB05A1 [10], která se nasadí přímo na vývojovou procesorovou desku. Tato přídavná deska komunikuje s procesorovou deskou pomocí SPI sběrnice.



Obrázek 8 – Modul Bluetooth LE typu X-Nucleo - IDB05A1 použitý pro komunikaci mezi ovládacím a promítacím zařízením

Dále může ovládací zařízení komunikovat s promítacím zařízením přes USB. Tato komunikace přes USB je určena zejména pro vývoj aplikace. Pro reálné použití není však příliš vhodná.

Komunikační pakety (bloky přenášených dat) jsou rozděleny na řídicí (nastavující stav zařízení) a průběhové (zadávaní průběhu pro pohyb stimulačního bodu). Řídicí pakety obsahují povel a případně doplňující data. Tyto pakety jsou následující (příklady paketů jsou uváděny jako posloupnost bajtů, tedy stylem „[bajt1, bajt2, ...]“, hranaté závorky symbolizují začátek a konec paketu):

- **Start** – Spuštění pohybu serv (podle nastaveného průběhu) a zapnutí laseru. [1]
- **Stop** – Zastavení pohybů serv a vypnutí laseru. [2]
- **Check** – Paket, který je odesílán každých 15 vteřin. Nepřijme-li promítací zařízení tento paket do 30 vteřin v průběhu rehabilitace, tak zastaví serva a vypne laser. [5]
- **ChangeDir** – Změna směru pohybu po trajektorii [6]
- **SpeedChange** – Změna rychlosti pohybu stimulačního bodu. [53, X]. X může nabývat celočíselných hodnot od 1 do 9 udávajících rychlost pohybu stimulačního bodu od nejpomalejší po nejrychlejší respektive.
- **LED** – Nastavení stavu modré LED [71, X]. X nabývá následujících hodnot:
  - 1 – Svítí (Do konce rehabilitace zbývá více jak 5 minut)
  - 2 – Bliká pomalu (Do konce rehabilitace zbývá méně jak 5 minut)
  - 3 – Bliká rychle (Do konce rehabilitace zbývá méně jak 1 minuta)
- **SetPos** – Přímé nastavení pozice stimulačního bodu na stěně. [61, X, Y]. X a Y reprezentují x a y (logické) souřadnice stim. bodu na stěně.

Průběhové pakety obsahují typ průběhu a po něm následují data / parametry tohoto typu průběhu. Průběhové pakety jsou následující (příklady paketů jsou uváděny jako posloupnost bajtů, tedy stylem „[bajt1, bajt2, ...]“):

- **RouteLines** – Paket nesoucí informace o trajektoriích typu Lines. [101, X1, Y1, X2, Y2, ..., Xn, Yn] Xi a Yi reprezentují x a y souřadnice bodů definující trajektorii.
- **RouteCurves** – Paket nesoucí informace o trajektoriích typu Curves. [102, C, X0, Y0, P1, P2] C reprezentuje typ křivky a podle typu křivky má paket dále tento tvar:
  - **Elipsa** – [102, 111, X0, Y0, P1, P2] X0 a Y0 reprezentují souřadnice středu elipsy. P1 představuje velikost hlavní poloosy elipsy a P2 udává velikost vedlejší poloosy elipsy.
  - **Lemniscate** (Ležatá osmička) - [102, 112, X0, Y0, P1] X0 a Y0 reprezentují souřadnice středu lemniscate. P1 představuje šířku lemniscate od středu ve vodorovné ose.
  - **Srdce** - [102, 113, X0, Y0, P1] X0 a Y0 reprezentují souřadnice středu srdce. P1 představuje velikost srdce.

Pakety pro komunikaci mezi ovládacím a promítacím zařízením přes USB se skládají ze start značky, přenášených hodnot a stop značky. Přenášené hodnoty jsou bajty zapsané ve formátu AsciiHex, tedy jako dva znaky. Při komunikaci přes Bluetooth-LE se uvedené posloupnosti bytes zapisují jako bloky dat přímo do servis / charakteristik vytvářených promítacím zařízením (přenášené bytes se tedy nikterak neupravují). Softwarový modul pro komunikaci přes USB a Bluetooth LE poskytl vedoucí práce.

## 5.7 Ovládací aplikace

Trenažér bude ovládaný pomocí aplikace na ovládacím zařízení (chytrý telefon či tablet). Aplikace bude multiplatformní a bude se skládat z na sebe navazujících stránek. Vyvíjena bude zejména pro Google/Android a MS/Windows. Zařízení s operačním systémem Android bude komunikovat pomocí Bluetooth-LE, počítač s operačním systémem Windows bude komunikovat pomocí USB (virtuální UART), případně rovněž Bluetooth-LE. Aplikace na Windows slouží zejména pro vývoj. Aplikace na zařízení s operačním systémem Android je určena pro primární způsob obsluhy trenažéru uživatelem.

Uživatel si může v aplikaci zapnout/vypnout hlasový výstup pro informaci o aktuálním stavu / činnosti aplikace, zobrazování záhlaví u jednotlivých stránek a navolit časový interval mezi změnou směru pohybu trajektorie. Dále se uživatel musí připojit k promítacímu zařízení přes Bluetooth-LE (případně přes USB pro Windows). Je-li uživatel připojen, může si zvolit trajektorii pro rehabilitaci.

V aplikaci si lze zvolit (a případně upravit) jeden ze 4 následujících typů trajektorií.

- **Přímky** (Lines) – Průběh se skládá z bodů. Tyto body jsou následně propojeny přímkami.
- **Křivky** (Curves) – Sofistikovanější křivky (elipsa, ležatá osmička, ...). Uživatel nastaví střed křivky a bod, který křivku definuje (její velikost/tvar).
- **Vlastní** (Custom) – Uživatel si nakreslí vlastní trajektorii.
- **OnLine režim** – Uživatel z ovládacího zařízení přímo hýbe s bodem promítaným na stěnu.

Po nastavení trajektorie, a jejím případné úpravě, může uživatel zahájit rehabilitaci.

## 6 Implementace řešení

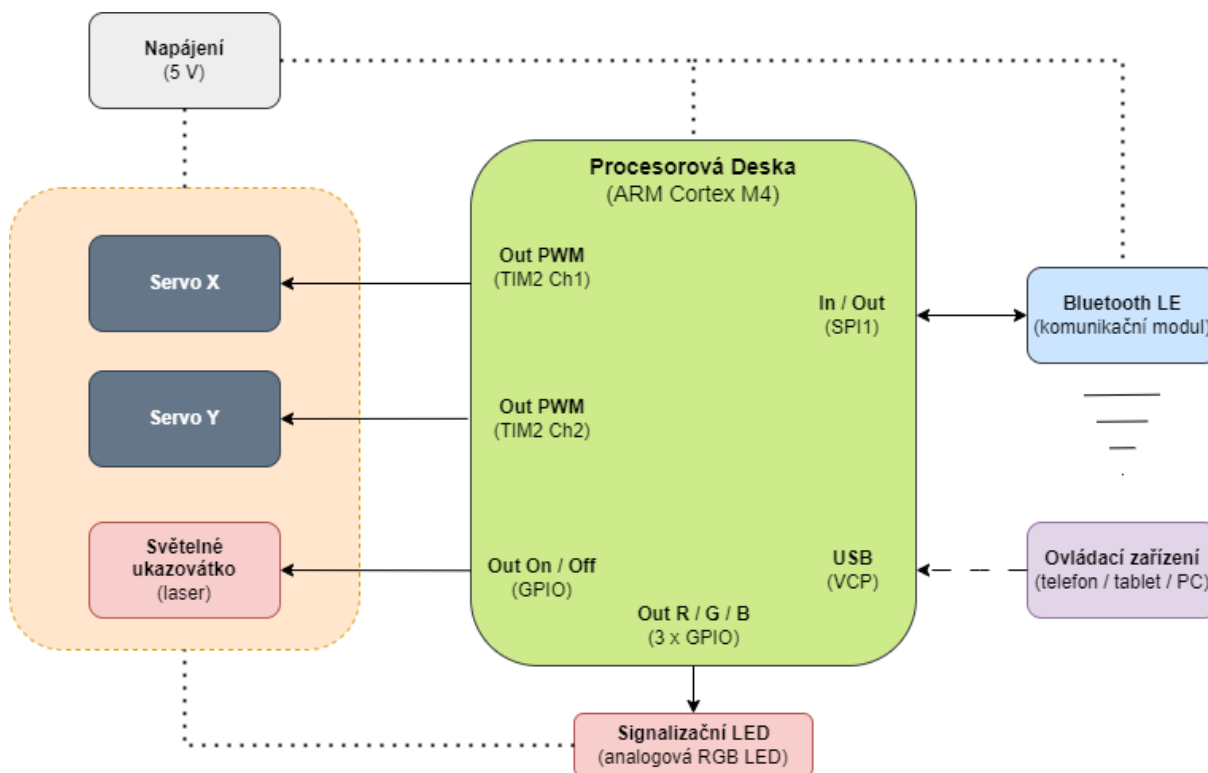
Pro programování procesorové desky s procesorem architektury ARM [6] jsem zvolil vývojové prostředí STM32cubeIDE [3],[7] a programovací jazyk C [2]. Pro tvorbu aplikace pro mobilní telefon / tablet s operačním systémem Android jsem vybral vývojové prostředí Visual Studio 2022 (Preview), kde je možné vyvíjet multiplatformní / mobilní aplikace pomocí .NET MAUI [8] v jazyce C# [1]. Při řešení problémů jsem využil informace získané z WWW stránek [5].

### 6.1 Promítací zařízení

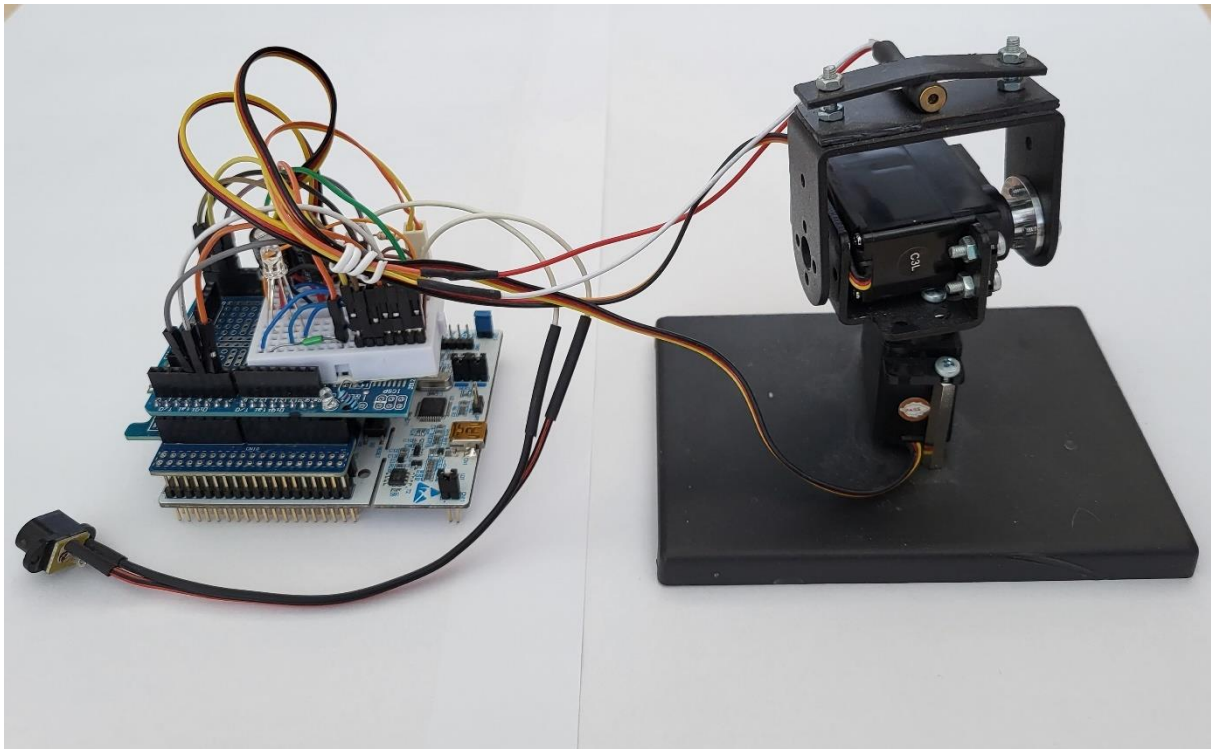
Jak vyplývá z návrhu, promítací zařízení se skládá z následujících komponent:

- **Procesorová deska** – Procesor řídí pohyb serv, laser, signalizační LED a komunikaci s ovládacím zařízením.
- **Serva** – Zajišťují pohyb laseru ve dvou osách X / Y a tím pohyb stimulačního bodu na stěně.
- **Laser** – Promítá stimulační bod na cílovou stěnu.
- **Signalizační LED** – Informuje uživatele o stavu promítacího zařízení a případně o zbývajícím času rehabilitace.
- **Komunikační Bluetooth LE modul** – Zajišťuje komunikaci přes Bluetooth LE s ovládacím zařízením.
- **Napájení** – Síťový napájecí adaptér poskytující napájecí napětí 5V (1.3A) pro jednotlivé komponenty promítacího zařízení.

Na obrázku 8 lze vidět schéma s bloky znázorňující jednotlivé komponenty a jejich vzájemné propojení.



Obrázek 9 – Blokové schéma promítacího zařízení znázorňující propojení jednotlivých komponent



Obrázek 10 - Prototyp trenážeru: serva s laserem (vpravo) a procesorová deska s komunikačním modulem (vlevo)

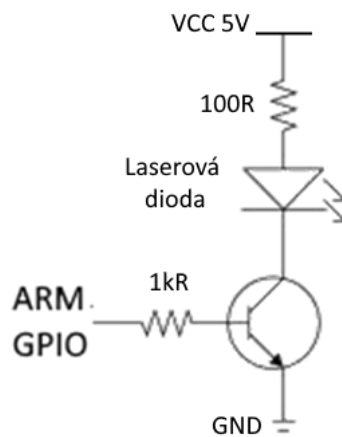
### 6.1.1 Ovládání pohonů

Jak již bylo zmíněno dříve, serva se ovládají pomocí PWM signálu. Konkrétně změnou střídy PWM signálu. Amplituda PWM signálu je hardwarově nastavena na 3.3 V (podle požadavku serv). Rovněž si nastavím potřebné výstupní piny na proc. desce tak, aby generovaly signál PWM o frekvenci 333 Hz potřebný pro řízení serv. To zajistí čítač/časovač (TIM2) nastavený na krok střídy 1/3600. Jelikož se natočení serv řídí nastavením střídy PWM signálu od 1/3 (max. výchylka jedním směrem) do 2/3 (max. výchylka druhým směrem), interval nastavení střídy je tedy zhruba [1200; 2400] z celkového rozsahu 3600.

Serva jsou připojena na řídicí PWM signál, na zem a na napájení 5 V.

### 6.1.2 Ovládání laseru

Laser se spíná pomocným tranzistorem typu NPN s označením 2N3904. Emitor spínacího tranzistoru je připojen na zem a kolektor v sérii s laserem a rezistorem na 5 V. Báze spínacího tranzistoru je připojena přes rezistor na pin vyvedený z procesorové desky.



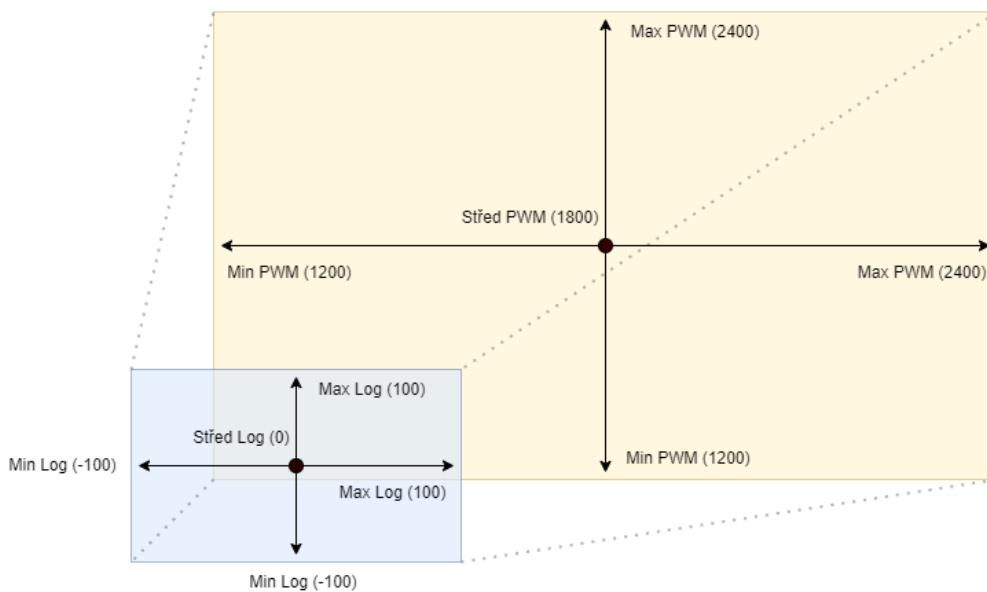
Obrázek 11 - Schéma spínání laserového modulu pomocí NPN 2N3904 tranzistoru

### 6.1.3 Signalizační LED

Jako signalizační LED je použita analogová RGB LED se společnou anodou. Signalizační LED má 4 piny. Jeden pin slouží k napájení LED (anoda) a zbylé 3 piny slouží k ovládání jednotlivých barev LED. Každý ze 3 pinů tedy řídí jednu ze základních RGB barev.

### 6.1.4 Trajektorie

**Souřadnice bodů v trajektorii** – V ovládací aplikaci se pro souřadnice bodů (průběhu) využívá vždy intervalu  $[-100; 100]$ , tedy Logické souřadnice. Tuto reprezentaci v ovládací aplikaci jsem zvolil, abych snížil objem dat přenášených v paketech, a aby rozsah zobrazeného pohybu / křivky nebyl závislý na velikosti / rozlišení displeje mobilního telefonu nebo tabletu. Procesorová deska však řídí serva pomocí PWM signálu s hodnotou střídy v intervalu  $[1200; 2400]$  (PWM hodnota  $\rightarrow$  výchylka serva  $\rightarrow$  pozice stimulačního bodu na cílové stěně = fyzické souřadnice). Tento interval je daný fyzickými možnostmi serv a řídicím PWM signálem. Kdybychom přenášeli souřadnice bodů jako přímo PWM hodnoty, zvětšily by se informace v přenášených paketech v podstatě dvakrát (logické = 1 bajt (8 bitů) / PWM hodnota = 2 bajty (16 bitů)). V promítacím zařízení tedy dochází k přepočtu ze souřadnic logických do souřadnic fyzických pro vystavení serv. Z obrázku 9 je zřejmý převod Logických souřadnic (modrá) na fyzické souřadnice / pozice na cílové stěně (s využitím PWM hodnot pro serva).



Obrázek 12 - Vztah mezi Logickým souřadnicemi (ovládací zařízení) a fyzickými souřadnicemi (cílová stěna)

Trajektorie jsou programově reprezentovány dvěma způsoby, pomocí struktur a parametricky.

**Reprezentování trajektorie pomocí struktur** – V kódu jsou vytvořeny dvě struktury s názvem Point a Trajectory. Tato reprezentace se týká trajektorií typu Lines a Custom.

```
//Struktura definující pozici bodu
typedef struct{
    uint16_t x; //Souřadnice X jako hodnota PWM [1200; 2400]
    uint16_t y; //Souřadnice Y jako hodnota PWM [1200; 2400]
} Point;

typedef struct{
    //Pole bodů obsahující trajektorii
    Point points[TRAJ_MAX_POINTS];
    uint16_t numOfPoints; //Počet bodů v aktuální trajektorii
    uint16_t currentPos; //Pozice bodu v trajektorii promítaného na stěnu
} Trajectory;
```

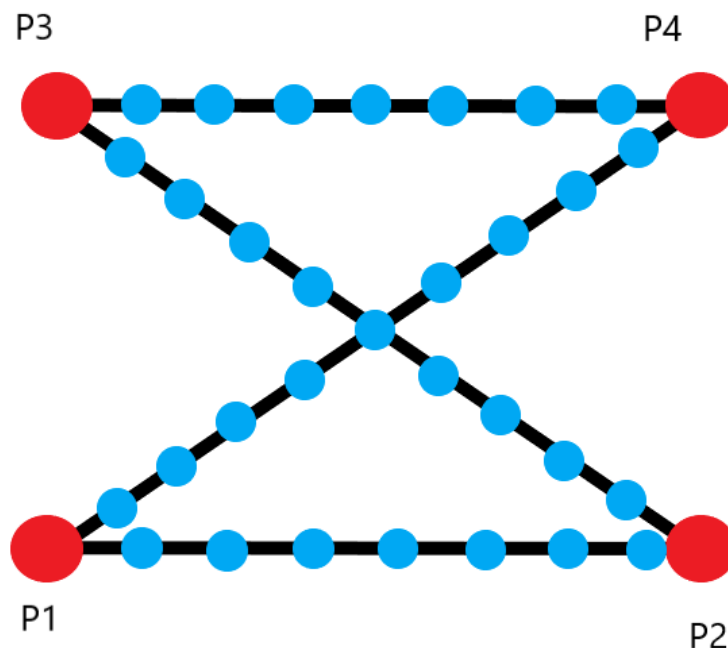
Jak lze vidět na předchozí ukázce kódu, strukturu Point tvoří dvě šestnácti-bitová čísla představující PWM hodnoty pro serva. Strukturu Trajectory tvoří pole bodů points a dvě šestnácti-bitová čísla reprezentující počet bodů v poli a pozici právě promítaného bodu na stěnu v aktuální trajektorii. Pohyb po trajektorii pak probíhá pomocí iterace bodů v poli points.

Výpočet probíhá tak, že z příchozího paketu, obsahujícího body definující trajektorii v logických souřadnicích, se tyto body přepočítají na PWM hodnoty pro serva. Následující pseudokód na obrázku 10 popisuje algoritmus výpočtu trajektorie (outputPoints) z bodů definující trajektorii (inputPoints).

```
vzdálenostBodů = délkaTrajektorie / početBodů; //vzdálenost mezi body v outputPoints
outputPoints[0] = inputPoints[0];
numOfOutputPoints = 1;
i = 0;
While(i < InputNumOfPoints){
    vzdálenostMeziBody //vzdálenost mezi body inputPoints[i] a inputPoints[i+1]
    vektorMeziBody //normalizovaný vector definovaný body inputPoints[i] a inputPoints[i+1]
    for (od 0 do vzdálenostMeziBody / vzdálenostBodů) {
        outputPoints[numOfPoints] = outputPoints[numOfOutputPoints - 1] + vektorMeziBody * vzdálenostBodů;
        numOfOutputPoints++;
    }
    i++;
}
```

Obrázek 13 - Pseudo kód pro výpočet kroků finální trajektorie typu Lines





Obrázek 14 – Princip doplnění bodů trajektorie přijaté z ovládacího zařízení (červené) o dílčí kroky (modré) pro jemný pohyb stimulačního bodu

Na obrázku 11 jsou vidět červené body určující (hrubě) trajektorii odeslanou z ovládacího zařízení a modré body doplněné promítacím zařízením, aby byl vhodně proložen zaslaný průběh pro plynulý pohyb stimulačního bodu po celé trajektorii.

**Reprezentování trajektorie parametricky** – Tato reprezentace se týká pouze trajektorií typu Curves. Tyto křivky jsou reprezentovány pomocí jejich parametrických předpisů. Předpisy pro křivky jsou následující:

- **Elipsa** – Elipsa je definována pomocí jejího středu  $(X_0, Y_0)$ , dvěma poloosami  $xAxis$  a  $yAxis$  a parametrem  $P$  (úhel). Její předpis je následující:

$$X = xAxis * \cos P + X_0 \quad (1)$$

$$Y = yAxis * \sin P + Y_0 \quad (2)$$

- **Lemniscate** (ležatá osmička) – Lemniscate je definován jeho středem  $(X_0, Y_0)$ , šířkou  $width$  a parametrem  $P$ . Jeho předpis je následující:

$$X = width * \frac{\cos P}{1 + \sin^2 P} + X_0 \quad (3)$$

$$Y = width * \frac{\sin P * \cos P}{1 + \sin^2 P} + Y_0 \quad (4)$$

- **Srdce** – Srdce je definováno jeho středem  $(X_0, Y_0)$ , velikostí  $size$  a parametrem  $P$ . Jeho předpis je následující:

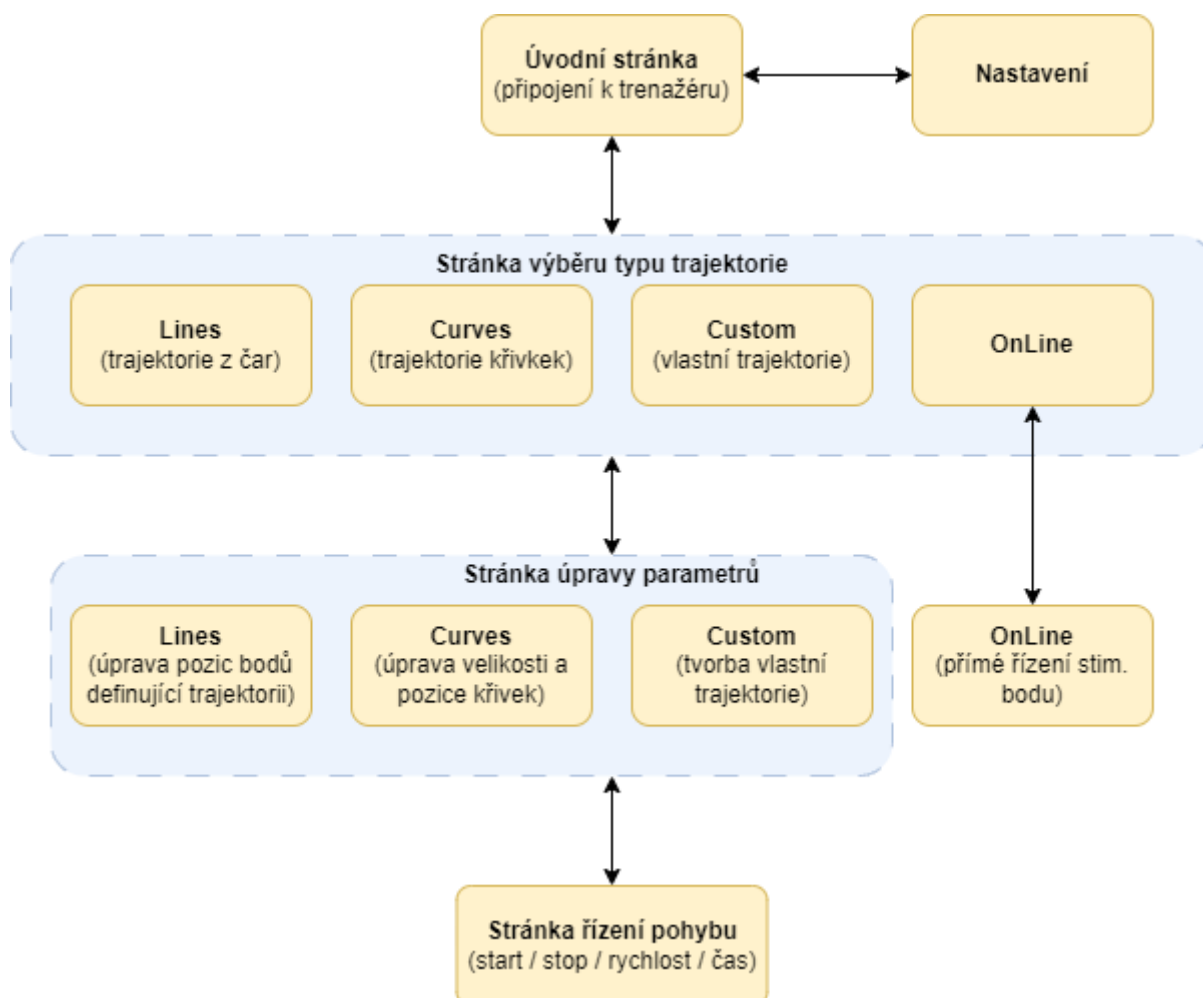
$$X = 16 * \sin^3 P * size + X_0 \quad (5)$$

$$Y = (13 * \cos P - 5 * \cos 2P - 2 * \cos 3P - \cos 4P) * size + Y_0 \quad (6)$$

Pohyb po trajektoriích reprezentovaných parametricky pak probíhá změnou parametru  $P$ .

## 6.2 Aplikace

Ovládací aplikace se skládá ze čtyř na sebe navazujících stránek a jedné stránky pro nastavení: úvodní stránka, stránka pro výběr typu trajektorie, stránka pro úpravu parametrů trajektorie a stránka řízení pohybu.



Obrázek 15 – Posloupnost použití stránek v ovládací aplikaci

**Úvodní stránka** obsahuje informaci, zda je uživatel připojen k trenažéru a případně k navázání tohoto spojení. Uživatel se z ní může dostat do stránky pro výběr průběhu nebo do stránky pro nastavení.

Stránka **Nastavení** – zde se vybírá Bluetooth-LE zařízení pro připojení (Android nebo Windows) či USB/VCP (Windows). Nastavují se zde rovněž některé zap/vyp proměnné jako např. hlasový výstup nebo zobrazení záhlaví stránek.

Stránka **Výběr typu křivky** slouží k výběru hlavního typu křivky a rovněž k výběru konkrétního průběhu. Přednastavené průběhy jsou uloženy jako soubory / obrázky na ovládacím zařízení. Vizualní obrázek má tedy čistě informační charakter pro uživatele, potřebné informace jsou pouze v jeho názvu. Přímě v názvu obrázku jsou tedy uvedeny body definující trajektorii například následovně (souřadnice X a Y jsou vždy odděleny čárkou):

“Ctverec (55,55)(55,-55)(-55,-55)(-55,55).png”

Takto pojmenovaný obrázek definuje svým názvem čtverec a je použit u průběhu typu Lines. Tento průběh obsahuje čtyři body spojené rovnou čarou. Pro extrakci pozic bodů z názvu obrázku jsem použil regulární výraz ve tvaru:

```
@("(?<posX>\-?\d+)[, ](?<posY>\-?\d+)")
```

Takovýto zápis extrahuje z názvu obrázku pouze body což je vhodné pro průběhy typu Lines a Curves. U průběhů typu Curves je v názvu současně přítomen i název křivky, jenž je důležitý pro identifikaci křivky a následný výpočet jejích bodů. Název takového obrázku vypadá následovně:

"Lemniscate (0,0)(85,0).png"

Obrázky samotné se pak zobrazí jako seznam na stránce ovládací aplikace a uživatel může vybrat požadovaný z nich pouhým dotykem. Uživatel tedy vybere obrázek a aplikace z jeho názvu vyzvedne potřebné parametry.

**Stránka úpravy parametrů** – zde se na plátno vykreslí průběh navolený na předešlé stránce. Plátno je ve formátu 16:9. Vertikální souřadnice je omezená na interval [-60; 60]. Tím se docílí formátu obrazce cca 16:9 i na cílové stěně.

V případě průběhu Lines se zobrazí červené body definované v názvu obrázku. Tyto body se spojí černou čarou reprezentující finální trajektorii.

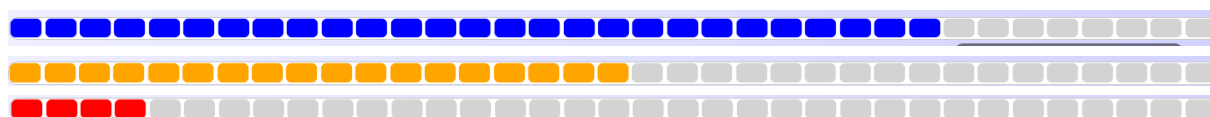
U průběhů typu Curves se zobrazí zvolená křivka a ji definující body. Zobrazená křivka je vypočítána stejnými rovnicemi jako u výpočtu průběhů v promítacím zařízení.

U průběhů typu Custom si zde uživatel nakreslí svůj vlastní průběh. Tento průběh je uložen v poli bodů a je upraven / zjednodušen tak, aby se odesílaný průběh skládal z maximálně 100 bodů (vysvětleno dále).

Při přechodu z této stránky na stránku pro Řízení pohybu, se odešle vybraný / vytvořený průběh do promítacího zařízení kde je dále zpracován.

Komunikaci mezi ovládacím zařízením a promítacím zařízením zajišťuje zdrojový soubor GlobalCommun.cs (dodaný vedoucím práce). V něm metoda Send() automaticky rozpozná jedná-li se o komunikaci přes Bluetooth-LE či USB/VCP a odešle paket(y) do promítacího zařízení.

**Stránka řízení pohybu** slouží k ovládání pohybu promítacího zařízení. Na pokyn uživatele se zde pomocí komunikační metody Send() odesílají povely pro pohyb promítacího zařízení. Rovněž je zde časová lišta indikující zbývající dobu rehabilitace (pohybu stimulačního bodu po trajektorii). Časová lišta se skládá z modrých obdélníků, které postupně šednou a tím indikují zbývající čas rehabilitace. Když do konce rehabilitace zbývá méně jak 5 minut, ještě aktivní obdélníky změní barvu z modré na oranžovou. Zbývá-li méně než 1 minuta do konce rehabilitace, ještě aktivní obdélníky změní barvu na červenou. Zastaví / přerušil-li se rehabilitace, všechny obdélníky v časové liště okamžitě zešednou. Ukázky časové lišty je možné vidět na obrázcích níže.



Obrázek 16 - Ukázky časové lišty: více než 5 minut (modrá), více než minuta (oranžová), méně než minuta (červená)

Délka trvání rehabilitace se měří pomocí časovače odčítajícího časové kroky 15s od nastavené celkové délky rehabilitace. Z toho plyne aktualizace časové lišty každých 15s. Tento časovač také zajišťuje odesílání paketů Check (kontrola spojení).

Aby se spustil a běžel pohyb laseru (i jeho svit), musí uživatel stále držet tlačítko start. Pustí-li jej, zastaví se z bezpečnostních důvodů serva a vypne se laser.

### 6.3 Tvorba a zpracování trajektorie

V této kapitole je celkem podrobněji popsán proces tvorby a zpracování jednotlivých typů trajektorií.

#### 6.3.1 Lines – čárový průběh

##### Ovládací zařízení

1. Uživatel zvolí typ trajektorie „Čáry“.
2. Ze složky s názvem „Lines“ v paměti na ovládacím zařízení se načtou obrázky obsahující před-připravené průběhy a ty se zobrazí jako seznam pod tlačítky pro zvolení typu průběhu.
3. Uživatel zvolí jeden z obrázků a zelenou šipkou přejde na následující stránku.
4. Při zvolení obrázku se metodou „ImageLinesSelected(name)“ z názvu zvoleného obrázku pomocí regulárního výrazu vyzvednou body definující trajektorii a uloží se do polí bodů points a pathPoints na ovládacím zařízení. Tzn. pro průběhy typu Lines jsou pole points a pathPoints identická.

```
183 public static void ImageLinesSelected(string name) { //gets points from the name of the file
184     string pattern = @"(?<posX>\-?\d+)[,](?<posY>\-?\d+)";
185     Regex rgxDevice = new Regex(pattern);
186     pathPoints.Clear();
187     points.Clear();
188     foreach (Match match in rgxDevice.Matches(name))
189     {
190         string posX = match.Groups["posX"].Value;
191         string posY = match.Groups["posY"].Value;
192         RoutePoint point = new Move.RoutePoint() { posX = int.Parse(posX), posY = int.Parse(posY) };
193         points.Add(point);
194         pathPoints.Add(point);
195     }
196 }
197
198 }
```

Obrázek 17 - Ukázka funkce "ImageLinesSelected()"

5. Na následující stránce se zobrazí červeně body z pole points a vykreslí se čáry propojující body v poli pathPoints.
6. S body v poli points lze přesouvat, a tím v podstatě upravovat zobrazenou trajektorii. Toto vše pomocí metod OnStartInteraction(), OnDragInteraction() a OnEndInteraction.
7. V metodě OnStartInteraction() se volá metoda FindClickedPoint(List pointList, int x, int y) detekující bod v poli points, se kterým uživatel chce hýbat. Pointer na tento bod se uloží do pracovní proměnné ClickedPoint.

8. V metodě OnDragInteraction() se aktualizují souřadnice bodu ClickedPoint a tím se bod posunuje na plátně.

```
25 public static RoutePoint FindClickedPoint(List<RoutePoint> pointList, int x, int y)
26 { //Finds point in pointList closest to (x,y)
27     foreach (RoutePoint point in pointList)
28     {
29         if (getDistPoint2Point(point, x, y) < Math.Pow(MoveConsts.dragDist, 2))
30         {
31             return point;
32         }
33     }
34     return null;
35 }
```

Obrázek 18 - Ukázka funkce "FindClickedPoint()"

9. V metodě OnEndInteraction(), volající se při puštění bodu, se nastaví pracovní pointer ClickedPoint na null. Pohyb bodem je ukončen.
10. Při přechodu na další stránku se pomocí metody FillSendMsg() zpracuje pole points a vytvoří se paket pro odeslání průběhu z ovládacího zařízení do promítacího zařízení.

#### Promítací zařízení

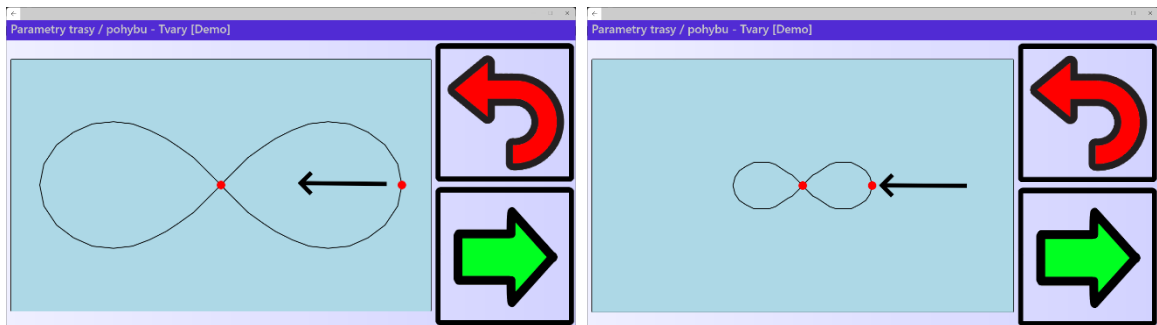
11. Z přijatého paketu se rozpozná průběh typu Lines, vyzvednou se všechny body definující tento průběh a uloží se do struktury BaseTrajectory.
12. Tyto body se pomocí metody GetTrajectory() proloží pro vytvoření dostatečně jemného pohybu a výsledná trajektorie se uloží do struktury Trajectory.
13. Nyní je možné procházet jednotlivé body průběhu a v podstatě je promítat na stěnu.

#### 6.3.2 Curves – parametrické křivky

##### Ovládací zařízení

1. Uživatel zvolí typ průběhu „Křivky“.
2. Ze složky s názvem „Curves“ v paměti na ovládacím zařízení se načtou obrázky obsahující předdefinované křivky a zobrazí se jako seznam pod tlačítka pro zvolení typu průběhu.
3. Uživatel zvolí jeden z obrázků.
4. Při výběru obrázku se pomocí metody „ImageCurvesSelected(name)“ určí o jakou křivku jde a pomocí regulárního výrazu se z názvu obrázku vyzvednou body křivku definující. Tyto body se uloží do pole points.
5. Z bodů v poli points se vypočítají parametry dané křivky a pomocí metod „ComputeEllipsePathPoints()“, „ComputeLemniscatePathPoints()“ nebo „ComputeHearthPathPoints()“ se do pole pathPoints uloží body představující křivku jako průběh.
6. Uživatel přejde na následující stránku. Na plátně se vykreslí červeně body definující křivku uložené v poli points a rovněž se vykreslí čáry propojující body v pathPoints. Takto vykreslené čáry zobrazují danou křivku.
7. Uživatel může opět přesouvat body v poli points (červeně zobrazené body) pomocí pointeru clickedPoint, metody „FindClickedPoint()“ a metod „OnStartInteraction()“, „OnDragInteraction()“ a „OnEndInteraction()“.

- Pohybem těchto červených bodů, definujících křivku, v poli points se automaticky pomocí metody „Compute...PathPoints()“ přepočítávají body v poli pathPoints a křivka se překresluje podle aktuálního nastavení / parametrů. Je-li uživatel s velikostí a pozicí křivky spokojen, může přejít na další stránku.



Obrázek 19 - Změna tvaru křivky před (vlevo) a po (vpravo) interakci s uživatelem

- Při přechodu na další stránku se body v poli points zpracují pomocí metody „FillSendMsg()“ a vytvoří se paket obsahující informace o typu průběhu, typu křivky a jejích parametrech. Tento paket je odeslán do promítacího zařízení.

#### Promítací zařízení

- V promítacím zařízení je paket zpracován. Rozpozná se o jakou křivku jde, rovněž se vyzvednou a uloží parametry křivky.
- Při promítání křivky na stěnu se pomocí rovnic a parametru P počítá pozice stimulačního bodu na křivce.

#### 6.3.3 Custom – vlastní definované křivky

##### Ovládací zařízení

- Uživatel zvolí typ křivky „Vlastní“ a zelenou šipkou přejde na další stránku.
- Na následující stránce uživatel nakreslí vlastní křivku, která se má promítat na stěnu.
- Kreslení probíhá pomocí metod „OnStartInteraction()“, kdy se uloží první bod tvořeného průběhu do pole pathPoints, „OnDragInteraction()“, kde se ukládají další body tvořeného průběhu rovněž do pole pathPoints tak, jak uživatel postupně kreslí na plátno a nakonec „OnEndInteraction()“, která uloží poslední bod průběhu.
- Uložených bodů definujících uživatelův průběh je většinou příliš mnoho a je potřeba jejich počet poněkud zredukovat. Toto zajišťuje metoda „GetFinalType3Route()“ kde se odeberou

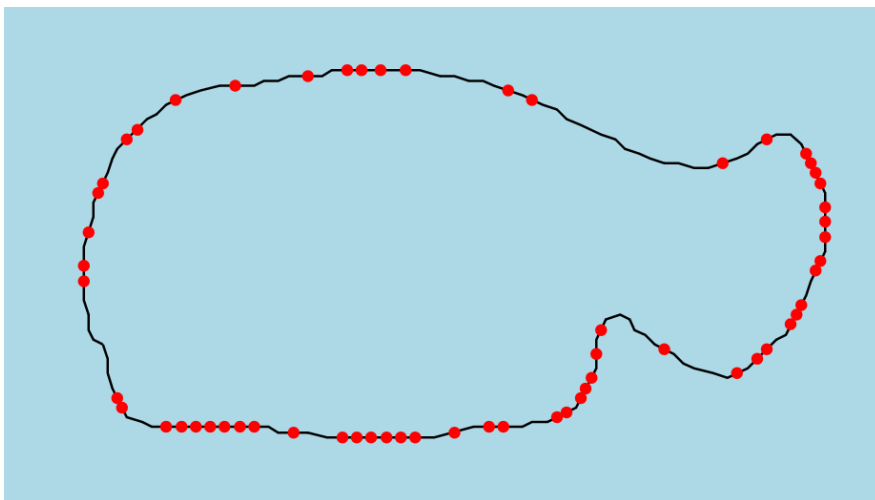
body z pole pathPoints ležící téměř na jedné přímce. Ukázka implementace této metody je na obrázku 18.

```

237 public static void GetFinalType3Route() {
238     int angleHranice = 2; //reduces amount of points drawn by the user under 100
239     points.Clear();
240     do
241     {
242         int i = 1;
243         while (i < pathPoints.Count() - 1)
244         {
245             int vect1X = pathPoints[i + 1].posX - pathPoints[i].posX;
246             int vect1Y = pathPoints[i + 1].posY - pathPoints[i].posY;
247             int vect2X = pathPoints[i].posX - pathPoints[i - 1].posX;
248             int vect2Y = pathPoints[i].posY - pathPoints[i - 1].posY;
249             double angle = getAngleVect(vect1X, vect1Y, vect2X, vect2Y);
250             if (angle <= angleHranice)
251             {
252                 pathPoints.RemoveAt(i);
253                 continue;
254             }
255             i += 1;
256         }
257         angleHranice += 1;
258     } while (pathPoints.Count() > 100);
259 }
260 }

```

Obrázek 21 - Ukázka funkce "GetFinalType3Route()"



Obrázek 20 – Příklad stejnolehých nepotřebných bodů odstraněných z vytvořené trajektorie typu Custom

Na obrázku 17 výše jsou červeně vyznačeny body v nakresleném průběhu, které byly z průběhu odstraněny metodou „GetFinalType3Route()“. Tyto body nepřidávaly do průběhu skoro žádné detaily a zbytečně by pouze prodlužovaly paket odesílaný do promítacího zařízení. Při (rychlejších) pohybu stimulačního bodu po stěně, by velmi pravděpodobně tyto detaily stejně nebyly postřehnutelné.

5. Z takto zpracovaných bodů se vytvoří pomocí metody „FillSendMsg()“ paket odesílaný do promítacího zařízení.

### Promítací zařízení

6. Tento paket je přijat ovládacím zařízením a je zpracován stejně, jako kdyby se jednalo o průběh typu „Lines“.

#### 6.3.4 OnLine – přímé řízení stimulačního bodu

1. Uživatel zvolí průběh typu „OnLine“ a může rovnou přejít na další stránku.

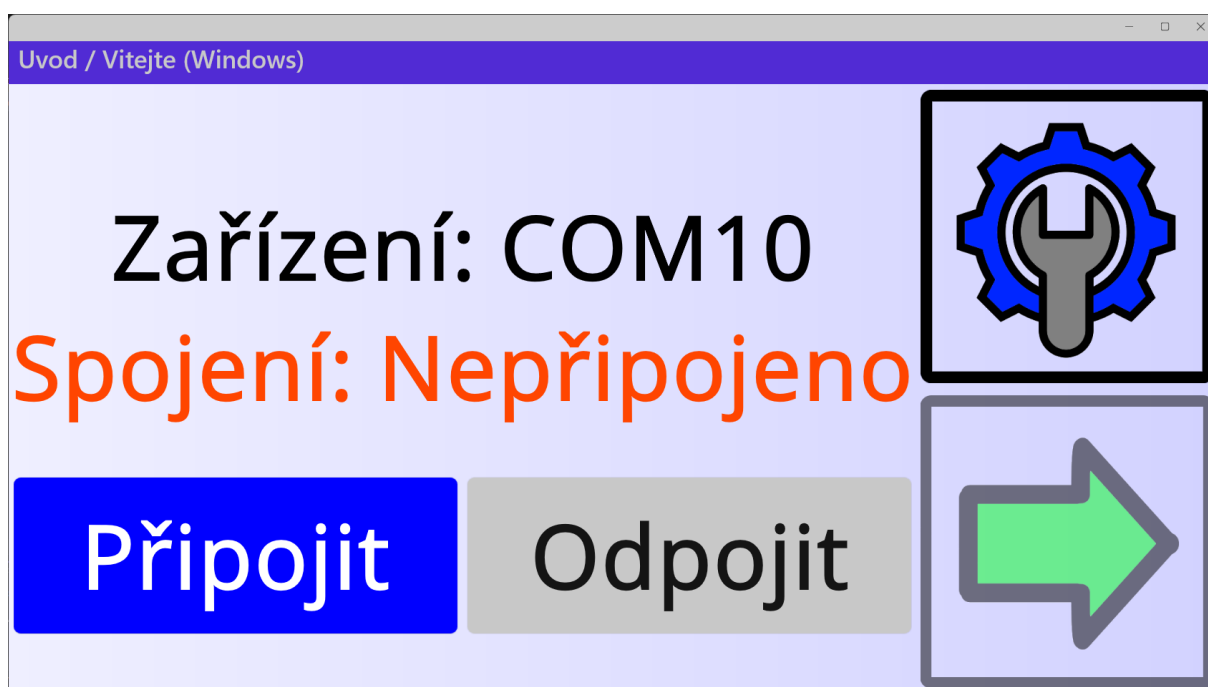
2. Na následující stránce se zobrazí plátno obsahující pouze jeden červený bod (v jeho středu). S tímto bodem může uživatel pohybovat.
3. Začne-li uživatel pohybovat s bodem po plátně, pomocí metody „OnStartInteraction()“, tak se rovněž zapne laser na promítacím zařízení.
4. Pohybuje-li uživatel s tímto bodem, metoda „OnDragInteraction()“ periodicky zasílá do promítacího zařízení paket s povelom setPosition obsahující aktuální souřadnice pohybujícího se bodu na plátně. Promítací zařízení tedy okamžitě nastavuje pozici stimulačního bodu na stěně podle přijatých souřadnic.
5. Přestane-li uživatel hýbat s bodem (pustí jej), pomocí metody „OnEndInteraction()“ se vypne laser.



## 7 Návod k použití

Uživatel na svém mobilním zařízení s Google/Android nainstaluje aplikaci (typu APK) a tu poté spustí (pomocí ikony). Zobrazí se informační obrazovka s odkazem na stáhnutí manuálu k použití. Dotykiem kamkoli na informační obrazovku (kromě odkazu na manuál) se uživatel přesune na úvodní stránku aplikace. Na úvodní stránce se nachází informace o vybraném / připojeném promítacím zařízení a čtyři tlačítka. Tedy tlačítko vedoucí na stránku pro nastavení, tlačítka pro připojení a odpojení od trenažéru / zařízení a tlačítko posouvající uživatele na další stránku. Popis stránky pro nastavení a proces připojení je uveden (až) na konci návodu k použití. Po připojení k trenažéru se zvýrazní šipka, umožňující přesun uživatele na další stránku. Při přechodu mezi stránkami je uživatel hlasovým výstupem informován o tom, na jaké stránce se nachází (je-li hlasový výstup povolen v nastavení aplikace).

Pokud není povoleno připojení k reálnému zařízení, tak se ovládací aplikace nachází v režimu „Demo“, což je textově signalizováno v záhlaví každé stránky aplikace.



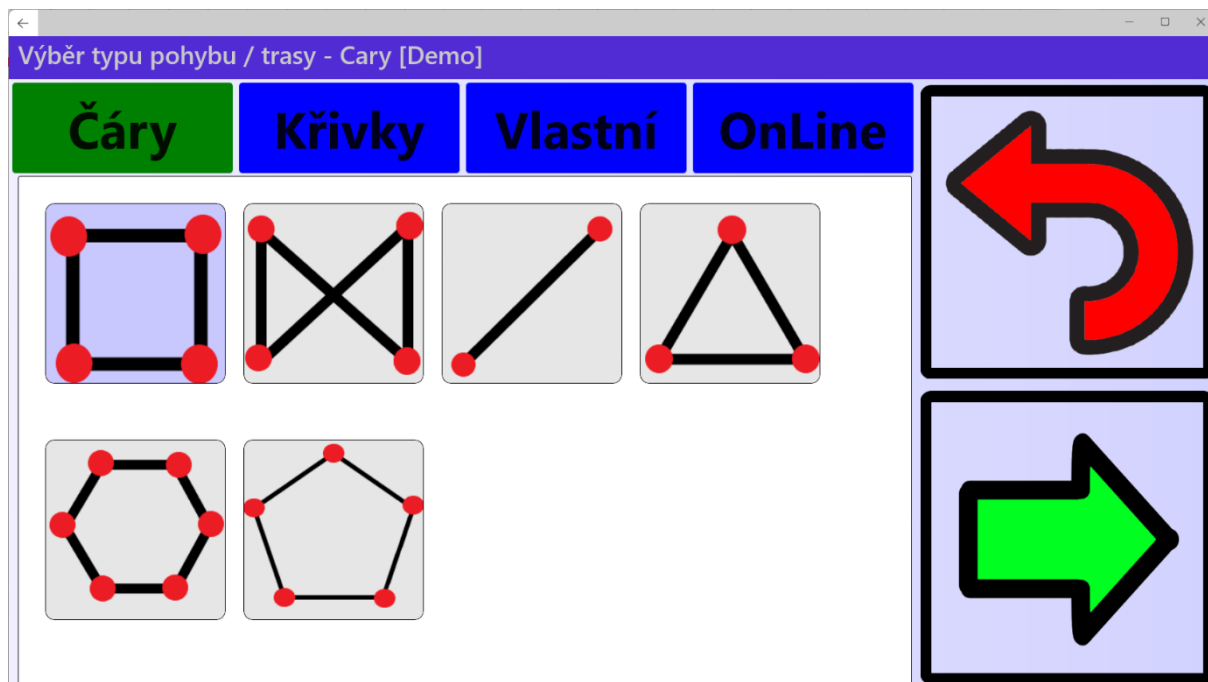
Obrázek 22 - Úvodní stránka ovládací aplikace s možností připojení k promítacímu zařízení

Nyní se uživatel nachází na stránce výběru průběhu. Zde si lze zvolit typ průběhu pomocí čtyř tlačítek s názvy „Čáry“, „Křivky“, „Vlastní“, „OnLine“, v horní části obrazovky. Při prvním příchodu na stránku je automaticky aktivní tlačítko „Čáry“ a v poli pod tlačítky se zobrazí obrázky s přednastavenými průběhy typu Čáry. Při zvolení jiného průběhu se změní barva daného tlačítka a o změně zvoleného průběhu ovládací zařízení informuje také hlasově (je-li povoleno).

Při výběru „Křivky“, stejně jako při stisku tlačítka „Čáry“, se v poli zobrazí přednastavené průběhy typu Křivky a to: „Elipsa“, „Ležatá osmička“, „Srdce“.

Vybral-li uživatel jeden z přednastavených průběhů typu Čáry nebo Křivky, může se přesunout na další stránku, kde lze zvolený přednastavený průběh dodatečně upravit podle vlastní potřeby.

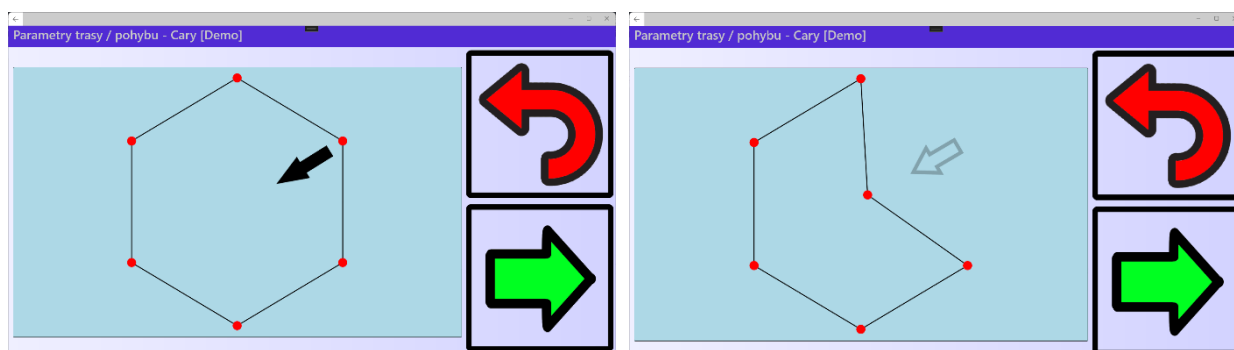
Zvolí-li uživatel zbylá dvě tlačítka „Vlastní“ nebo „OnLine“, může se rovnou přesunout pomocí zelené šipky na další stránku pro tvorbu průběhu nebo na stránku, kde se přímo ovládá pohyb promítaného stimulačního bodu.



Obrázek 23 – Stránka pro výběr typu průběhu: „Čáry“ (vybráno), „Křivky“, „Vlastní“ a „OnLine“

Stránka pro úpravu parametrů průběhu se liší podle typu vybraného průběhu. V horní liště se zobrazí typ zvoleného průběhu a uživatel je o něm také hlasově informován (je-li povoleno v nastavení). Na této stránce lze vidět modrý obdélník (plátno), na kterém se zobrazují zvolené průběhy, a současně představuje rozsah pohybu stimulačního bodu na stěně.

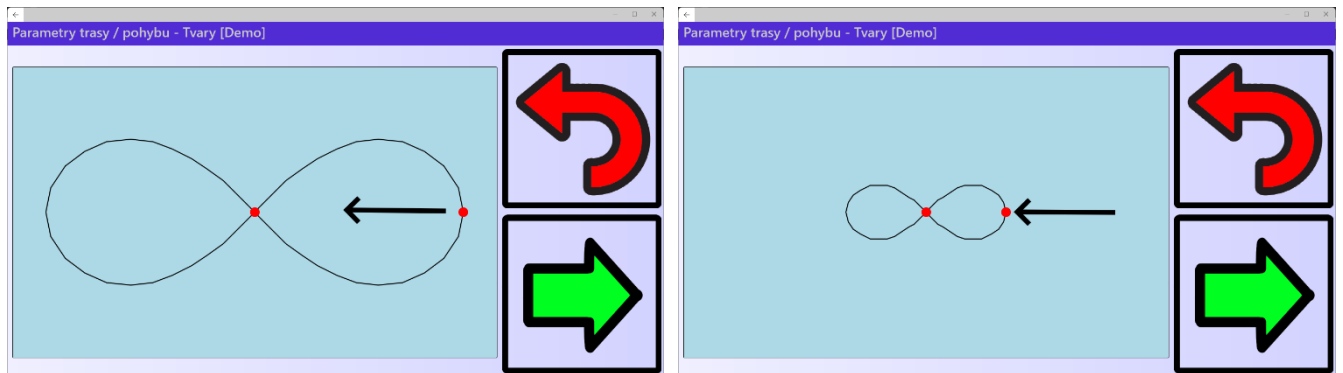
Zvolil-li uživatel typ průběhu „Čáry“, zobrazí se na plátně průběh navolený pomocí obrázku na předešlé straně. Uživatel může trajektorii pro pohyb stimulačního bodu upravit přesunutím jakéhokoli z červených bodů na plátně.



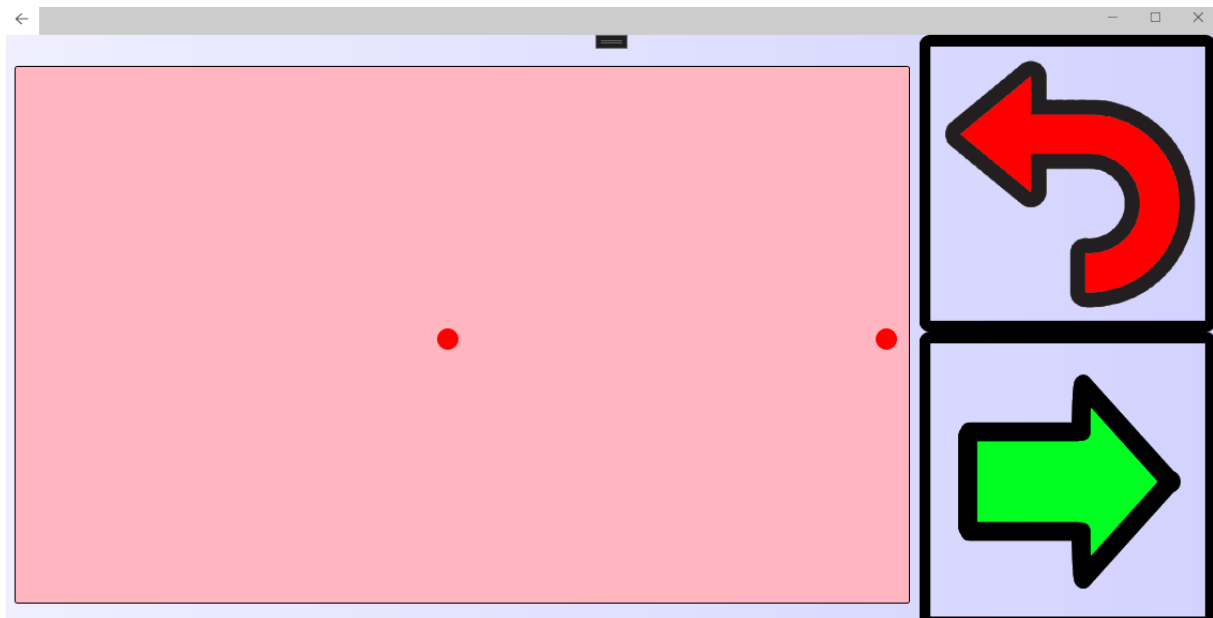
Obrázek 24 – Stránka pro úpravu bodů trajektorie typu „Čáry“- před úpravou (vlevo) a po úpravě (vpravo)

Zvolil-li uživatel typ průběhu „Křivky“, zobrazí se na plátně rovněž zvolená křivka, ale v tomto případě pouze dva body pro její úpravu. Uživatel může s těmito body pohybovat a současně sledovat, jak se křivka mění. Je-li jakýkoli bod křivky mimo plátno, tak se změní barva pozadí na červeno-růžovou, křivka zmizí a není možné přejít na další stránku. Křivka musí být vždy zobrazena celá v rozsahu plátna. Je-li

uživatel s tvarem křivky spokojen, může se pomocí zelené šipky přesunout na další stránku pro řízení pohybu.

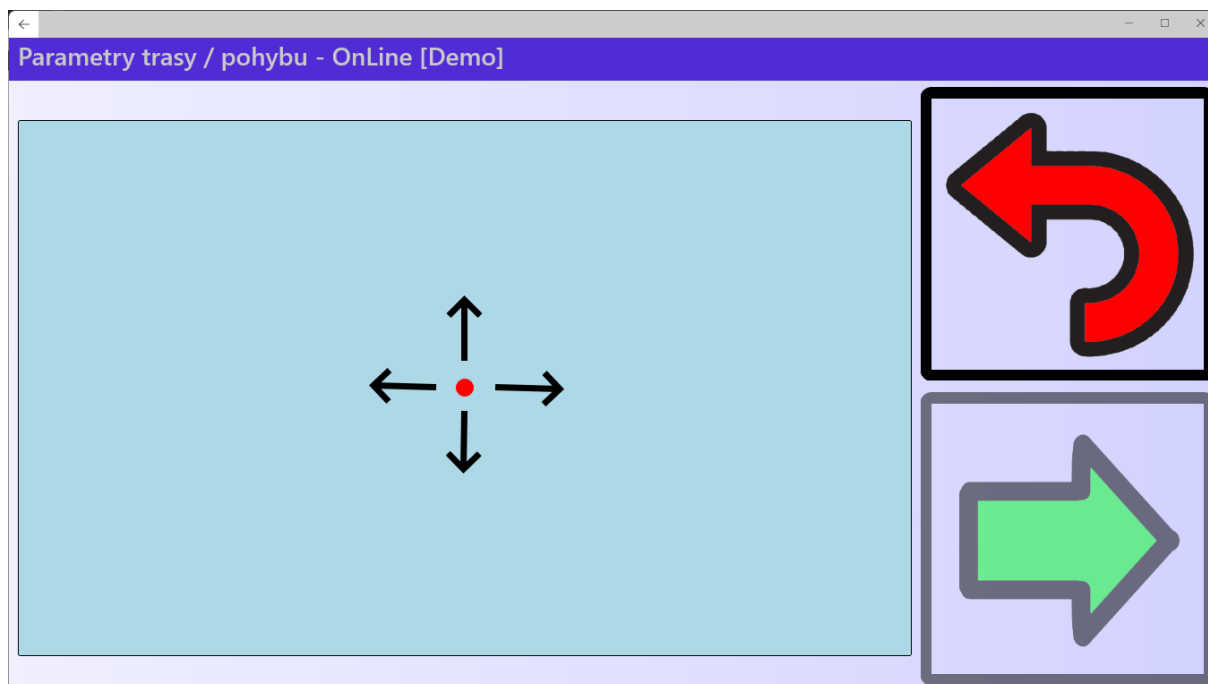


Obrázek 25 – Stránka pro úpravu parametrů trajektorie typu „Křivky (Lemniscate) – před úpravou (vlevo) a po úpravě (vpravo)“



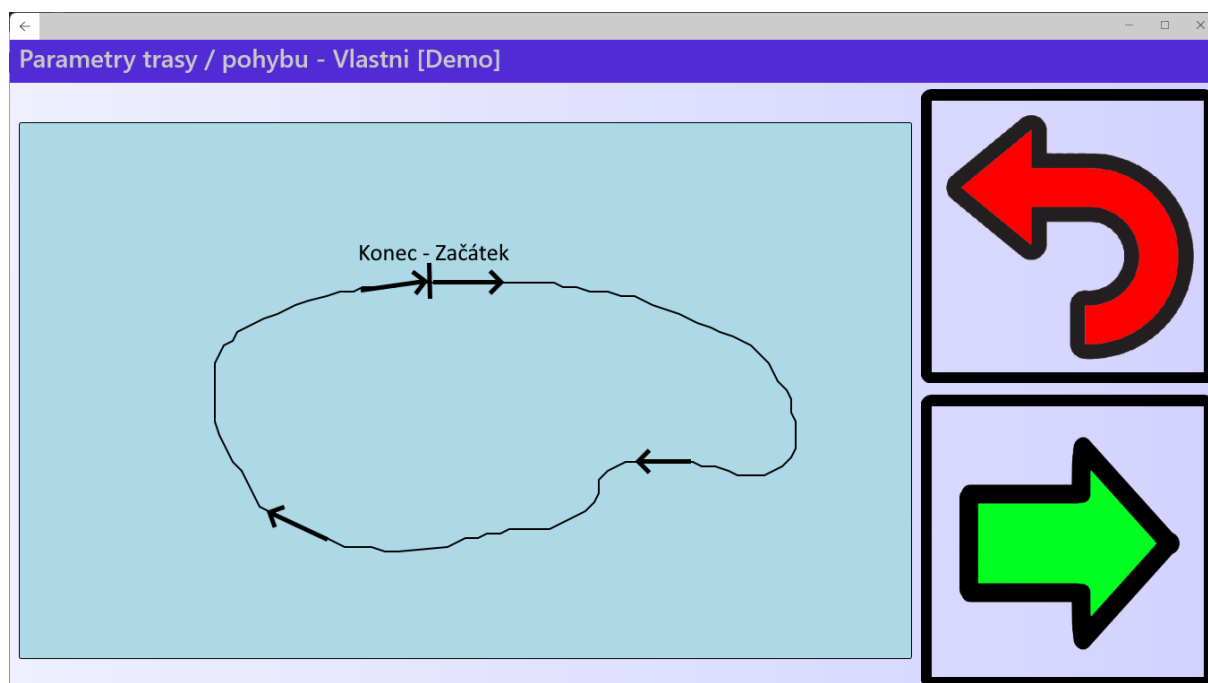
Obrázek 26 – Příklad signalizace (červeno-růžové pozadí) chybně nastavených parametrů křivky

Zvolil-li si uživatel typ průběhu „OnLine“, uprostřed plátna se nachází pouze jeden červený bod, se kterým lze pohybovat. Jakmile začne uživatel s bodem hýbat, zapne se laser a serva pohybují se stimulačním bodem na cílové stěně stejně, jako uživatel pohybuje s bodem na plátně ovládacího zařízení.



Obrázek 27 - Stránka pro přímé řízení pohybu stimulačního bodu na stěně („OnLine“)

Zvolil-li uživatel typ průběhu „Vlastní“, lze na stránce pro úpravu průběhu opět vidět plátno. Na toto plátno lze dotykem a tažením přes plátno nakreslit jakoukoli vlastní trajektorii pro pohyb stimulačního bodu. Jakmile se přestane kreslit na plátno, trajektorie se uzavře / upraví a lze vidět její finální podobu. Pomocí zelené šipky se uživatel přesune na další stránku pro řízení pohybu.



Obrázek 28 – Příklad uživatelem nakresleného průběhu typu „Vlastní“

Na stránce pro řízení pohybu se nachází následující tlačítka:

- Tlačítka **nastavení celkového času rehabilitace** – V horní části stránky se nachází 5 tlačítek pro nastavení doby trvání rehabilitace. Automaticky je nastaveno 15 minut. Změní-li uživatel délku doby rehabilitace stisknutím příslušného tlačítka, stisknuté tlačítko změní barvu.
- Tlačítko **Spuštění / Zastavení pohybu trenážeru** – Drží-li uživatel toto tlačítko, stimulační bod se pohybuje po zadané trajektorii a laser je rozsvícen. Stimulační bod se pohybuje po zadané trajektorii pouze a po celou dobu držení, tedy až do okamžiku puštění, tohoto tlačítka. Vykonávání pohybu po trajektorii je signalizováno modrou barvou LED. Puštěním tohoto tlačítka změní signalizační LED barvu z modré zpět na zelenou.
- Tlačítka pro **snížení a zvýšení rychlosti** – Stiskem těchto tlačítek se zvýší/sníží rychlost pohybu stimulačního bodu po trajektorii.
- Tlačítko pro **otočení směru pohybu** stimulačního bodu po trajektorii (pohyb opačným směrem).
- Tlačítko **zpět** – Zobrazí uživateli předchozí stránku. Toto tlačítko je šedé probíhá-li pohyb stimulačního bodu.

O stisku většiny tlačítek je uživatel rovněž hlasově informován (je-li povoleno).

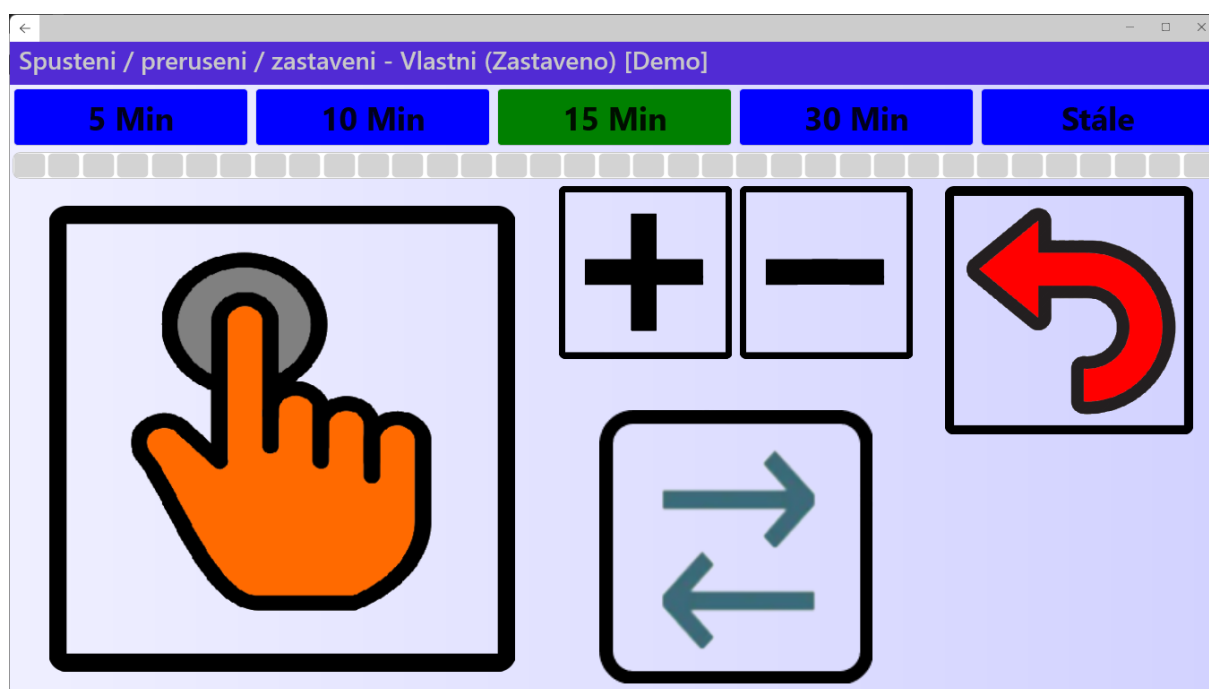
V horní části stránky pod tlačítka pro nastavení času se nachází časová lišta indikující zbývající čas rehabilitace. Lišta může mít 3 barvy indikující zbývající čas do konce rehabilitace:

- **Modrá** – Do konce rehabilitace zbývá více než 5 minut.
- **Oranžová** – Do konce rehabilitace zbývá méně než 5 minut.
- **Červená** – Do konce rehabilitace zbývá méně než 1 minuta.

Zároveň je uživateli indikován čas do konce rehabilitace pomocí signalizační LED na promítacím zařízení.

- **Svítí modře** - Do konce rehabilitace zbývá více než 5 minut.
- **Bliká modře pomalu** - Do konce rehabilitace zbývá méně než 5 minut.
- **Bliká modře rychle** - Do konce rehabilitace zbývá méně než 1 minuta.

Uživatele je také hlasově informován o zbývajícím čase (je-li povoleno).



Obrázek 29 - Stránka pro řízení (spuštění / zastavení / rychlost / atd.) pohybu stimulačního bodu na stěně

**Nastavení trenažéru a proces spojení přes Bluetooth** – Na úvodní stránce se uživatel pomocí tlačítka pro nastavení přepne na stránku pro nastavením. Zde lze nastavit mnoho parametrů pro chování aplikace:

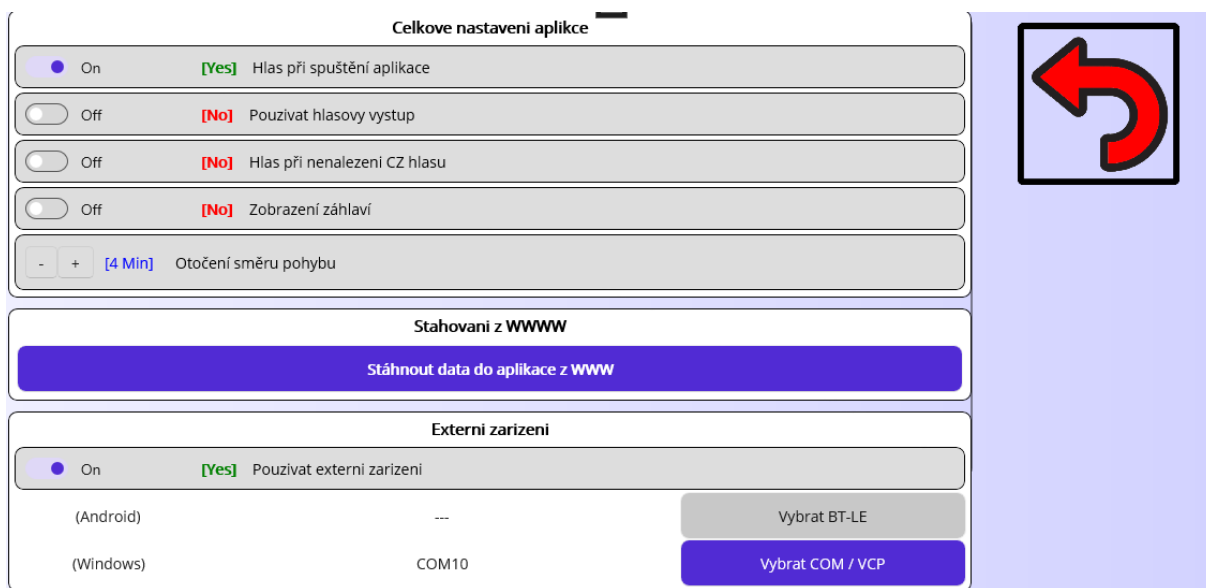
Zapnutí/vypnutí:

- Hlasového výstupu aplikace
- Zobrazení záhlaví stránek
- Hlas při zapnutí aplikace
- Varování při nenalezení českého převodu textu na řeč

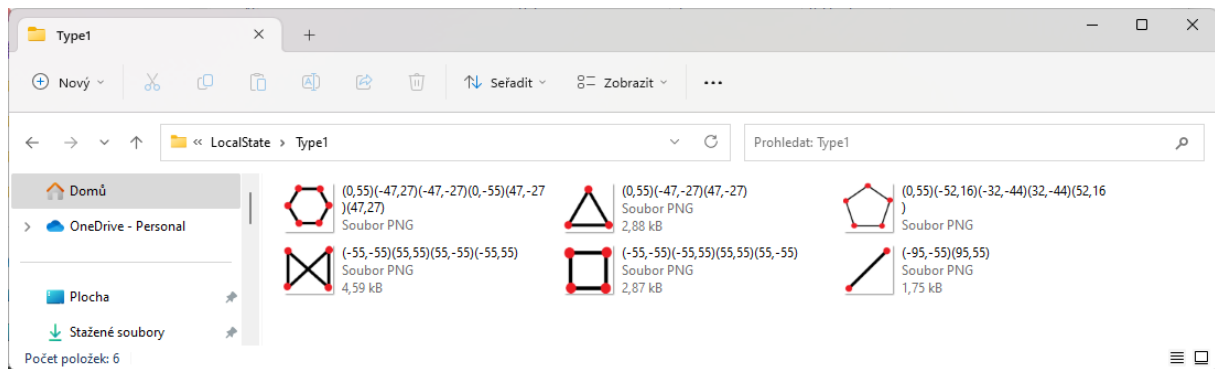
Dále si zde může uživatel nastavit dobu, po které se automaticky otočí směr pohybu po trajektorii (proti zamotání hlavy).

Při prvním spuštění aplikace si uživatel pomocí tlačítka „Stáhnout data do aplikace z WWW“ stáhne přednastavené průběhy z internetu do svého ovládacího zařízení.

Rovněž je nutno, aby se uživatel připojil k trenažéru. Aby se mohl uživatel připojit přes Bluetooth-LE, musí být zařízení trenažéru připojené do elektrické sítě a musí být zapnuté. Pokud je trenažér pouze zapnutý svítí signalizační LED červeně. Uživatel zmáčkne tlačítko „Vybrat BT-LE“ a vyhledá dostupná zařízení (oční trenažéry) v okolí. Uživatel vybere trenažér se jménem napsaném na štítku na krabici trenažéru (například „OcniTren001“). Po vybrání BT-LE zařízení, tedy požadovaného trenažéru, se může uživatel přesunout zpět na Úvodní stránku. Na úvodní stránce stiskne tlačítko „Připojit“. Při úspěšném spojení přes Bluetooth-LE se změní barva signalizační LED na trenažéru z červené na zelenou.



Obrázek 30 – Stránka pro nastavení činnosti aplikace a výběr promítacího zařízení



Obrázek 31 - Adresář pro uložení přednastavených průběhu typu "Lines"

Na obrázku 27 výše je vidět složka s uloženými průběhy typu Přímky. Jde tedy o soubory typu „obrázek“ obsahující ve svém názvu informace potřebné pro vytvoření tohoto typu průběhu.

## 8 Závěr

Cílem práce bylo vytvořit prototyp zařízení pro trénink okohybných svalů. Oproti současnému stavu, kdy jsou jiné trenažéry okohybných svalů buď nedostupné nebo velké a velice drahé, je můj trenažér relativně malý a cenově dostupný. Jeho velkou výhodou je možnost volby ze čtyř typů průběhů a možnost jejich případné úpravy. Pacient se tak může zaměřit na cvičení konkrétních okohybných svalů. Lze vybírat z následujících průběhů:

- **Přímky** – Pacient nastaví body, které se propojí přímkou a tím vznikne trajektorie pro pohyb stimulačního bodu na stěně.
- **Křivky** – Pacient si může vybrat (zatím) ze 3 typů křivek a nastavit jejich vlastnosti. Stimulační bod pak na stěně kopíruje nastavenou křivku.
- **Vlastní** – Pacient si vytvoří zcela vlastní trajektorii pohybu stimulačního bodu na stěně.
- **OnLine** – Pacient přímo ovládá pozici stimulačního bodu na stěně.

Dále je zde možnost si doplnit vlastní průběhy typu Přímky a uložit je pro další použití.

Tento trenažér odstraňuje některé hlavní nedostatky současného stavu jako např.:

- **Velikost** – Trenažér je relativně malý a tudíž se dá lehce uskladnit či přemístit.
- **Cena** – Je v porovnání s jinými současnými trenažéry levný a tím se rehabilitace okohybných svalů může stát dostupnější.
- **Jednoduše obsluhovatelný** – Trenažér je jednoduchý na obsluhu, tudíž se dá zapůjčit pacientovi, aby mohl rehabilitovat doma a nemusel často docházet do ordinace lékaře.

Kromě již zmíněných značných výhod oproti existujícím zařízením tohoto typu má zde popisované zařízení, také i některé nevýhody:

- Pro tvorbu stimulačního bodu je použit laser. I když je použitý laser velmi malého výkonu (svitu), tak přesto existuje nenulová pravděpodobnost, kdy si jím může uživatelem nepříjemně posvítit do oka / očí. Tato skutečnost může být pro lékaře / uživatele poněkud obávaná. Zařízení je však koncipováno tak, aby se tato pravděpodobnost skutečně minimalizovala (zejména způsobem ovládání celého zařízení).
- Další a asi nezávažnější nevýhodou oproti současným řešením je absence (zejména medicínské) certifikace zde popisovaného zařízení. Vytvořené zařízení je tedy (zatím) vhodné pouze pro testovací / experimentální / prototypové účely, nikoli běžné použití. Certifikace takového zařízení může být velmi nákladná a tím se může stát zařízení v podstatě neprodejně.

I když je vytvořené zařízení pouze ve stádiu prototypu, tak se jedná o velmi jednoduché zařízení se skutečně velkými možnostmi. Jeho skutečně velkou výhodou je tvorba v podstatě libovolné trajektorii pro pohyb stimulačního bodu.



## 9 Zdroje

- [1] JUAREZ, Rodrigo. NET MAUI for C# Developers. Packt Publishing, ISBN 1837631697
- [2] KERNIGHAN, Brian W. a Dennis M. RITCHIE. *Programovací jazyk C*. 2. vydání. Přeložil Zbyněk ŠÁVA. Brno: Computer Press, 2019. ISBN 978-80-251-4965-2.
- [3] IBRAHIM, Dogan. *Nucleo boards programming with the STM32CubeIDE: Hands-on in more than 50 projects*. Susteren: Elektor, 2020. ISBN 978-3-89576-416-5.
- [4] <http://cyber.felk.cvut.cz/novakpe/Projekty/Strabismus/PhpXmlTextToHtml.php?fileTexts=TextsPageHtmStrabEnter> - 2023
- [5] <https://stackoverflow.com/> - Pro ladění projektu a hledání řešení některých chyb v programech (.NET, STM32, C). - 2023
- [6] NOVIELLO, Carmine. *Mastering STM32*, LeanPub, 2023 ([www.carminenoviello.com](http://www.carminenoviello.com))
- [7] PAKDEL, Majid. *Advanced Programming with STM32 Microcontrollers*, Elektor, 2020, ISBN 978-3-89576-410-3
- [8] 1. PRICE, Mark J. *C 8. 0 and .NET core 3. 0-modern cross-platform development: Build applications with C, .NET core, entity Framework Core, asp.NET Core, and ML.net using visual studio code*. Birmingham, England : Packt Publishing, 2019.
- [9] <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html> - 2023
- [10] <https://www.st.com/en/ecosystems/x-nucleo-idb05a1.html> - 2023