



**Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Koterovská 85**

ROČNÍKOVÁ PRÁCE

Téma: Programovatelné auto

Autor práce: Jan Ocelík
Obor studia: 78-42-M/01 Technické lyceum
Třída: 3. L
Předmět: Kybernetika
Zadávající učitel: Jiří Švihla
Dne: 28. 4. 2023

Hodnocení:



**Vyšší odborná škola a
Střední průmyslová škola elektrotechnická
Plzeň. Kotterovská 85**

ZADÁNÍ ROČNÍKOVÉ PRÁCE	
Školní rok	2022/ 2023
Studijní obor	78-42-M/01 Technické lyceum
Jméno a příjmení	Jan Ocelík
Třída	3.L
Předmět	Kybernetika
Hodnoceno v předmětu	Kybernetika
Téma	Programovatelné auto
Obsah práce	<ol style="list-style-type: none">1. Sehnání součástek2. Design a tisk modelu (šasi auta)3. Sestavení prototypu4. Vytvoření Framework/SDK/API5. Naprogramování
Zadávající učitel Příjmení, jméno	Švhla Jiří
Podpis zadávajícího učitele	
Termín odevzdání	28. dubna 2023

Anotace

Cílem této ročníkové práce je za pomoci opensource programů navrhnut a z opensource komponent poskládat robotické autíčko pro začátečníky i pokročilé s možností jednoduchého sestavení. Dalším úkolem je vyvinout firmware a API pro jednoduché skriptové programování i komplexnější programování s využitím obrazového vstupu. Dalším úkolem je připravit ukázkové příklady kódu k předvedení jednotlivých funkcí robota. Posledním úkolem je zpříjemnit práci s robotem, zhodnotit přínosy a možnosti využití projektu ve vzdělávání a vypracovat potřebnou dokumentaci.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

Plzni dne: Podpis:

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Jiřímu Švihlovi za pomoc s výběrem komponent a obsahem dokumentace a rodině a přátelům za psychickou podporu. Také děkuji všem dohromady za to, že mě k tomu včas dokopali.

Obsah

Anotace	2
Poděkování	3
Obsah	4
1 Úvod	5
1.1 Cíle a požadavky	6
1.1.1 Komponenty (moduly)	6
1.1.2 Šasi (tělo)	6
1.1.3 API	6
1.2 Návrhový software	7
1.2.1 Onshape	7
1.2.2 Code OSS/VS Code	9
1.3 Výrobní technologie	11
1.3.1 3D tisk	11
2 Popis konstrukce a vybrané moduly	12
2.1 Vybrané moduly	12
2.2 Konstrukce	14
3 Software	16
4 Závěr	17
Seznam obrázků	20
Seznam příloh	21

1 Úvod

V dnešní době se stále více hovoří o automatizaci a digitalizaci a tyto trendy mají velký vliv na společnost. Robotika a autonomní systémy jsou jednou z klíčových oblastí, které se rozvíjejí v rámci těchto trendů a mají potenciál změnit mnoho aspektů našeho života.

Vzdělávání a výuka v této oblasti se také stávají stále důležitějšími, protože mnoho pracovních pozic, které budou v budoucnosti vyžadovat znalosti robotiky a programování, ještě neexistuje. Výuka v této oblasti tak může být klíčová pro přípravu studentů na pracovní trh budoucnosti.

Proto jsem se rozhodl věnovat svou ročníkovou práci právě problematice robotiky a vytvořit autonomní auto, které bude sloužit jako výuková pomůcka. Můj záměr je ukázat, jak moderní technologie mohou být využity k tomu, aby byli studenti lépe připraveni na budoucí výzvy v oblasti robotiky a informatiky.

Při tvorbě prototypu se budu snažit využít nejnovější poznatky v oblasti robotiky a programování, a to jak z teoretického hlediska, tak praktických zkušeností z mých předchozích aktivit. Cílem bude vytvořit zařízení, které bude snadno ovladatelné a srozumitelné pro studenty různých věkových kategorií, a zároveň bude dostatečně funkční a výkonné pro splnění zadaných úkolů.

Věřím, že tato ročníková práce přispěje k popularizaci robotiky a podpoří zájem studentů o tuto oblast. To může mít pozitivní vliv na jejich budoucí kariéru a na rozvoj robotiky v České republice.

1.1 Cíle a požadavky

1.1.1 Komponenty (moduly)

Hlavním cílem této ročníkové práce je návrh a výroba robota z běžně dostupných materiálů, v tomto případě ze snadno sehnatelných modulů.

Moduly by měly být zároveň opensource, aby si software pro ně mohl programovat každý, který se k robotovi dostane, nebo si ho sám vyrobí.

Důležitý je také výběr napájení, v případě robotického auta tedy baterie, tak, aby udrželo robota v provozu po přiměřeně dlouhou dobu a aby zároveň dokázalo vykryt proudové špičky.

1.1.2 Šasi (tělo)

Tělo robota by mělo být vyrobeno tak, aby bylo plně uzavíratelné, tedy bezpečné pro jakýkoli přenos i při nešetrném zacházení, ale aby se dal robot také provozovat s plně přístupnými veškerými propojkami a bylo tedy snadné ladit jeho software za provozu.

1.1.3 API

Jedním z dalších požadavků je co nejvíce zpříjemnit uživateli práci s robotickým autem, tedy vytvořit nějaké API nebo knihovnu.

Nedílnou součástí je také dokumentace k danému API/knihovně a to proto, aby si i nový uživatel zvládl bez jakékoli pomoci robota pomocí daného API/knihovny naprogramovat.

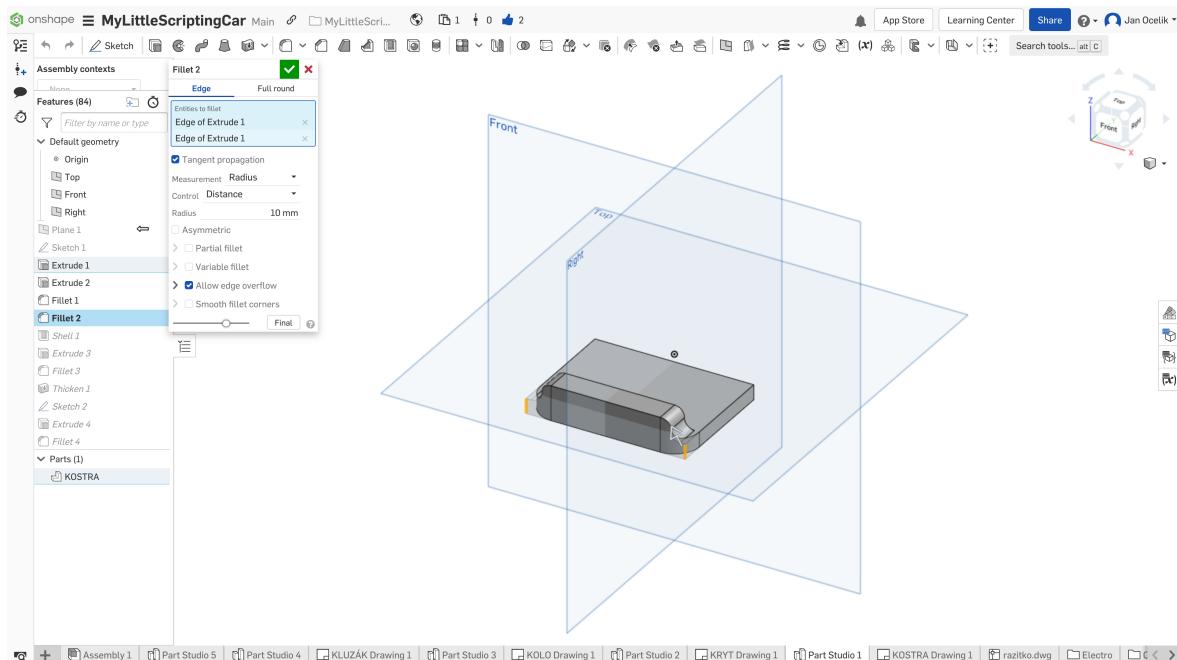
1.2 Návrhový software

1.2.1 Onshape

Onshape je online software pro návrh 3D součástí. V programu se vytváří takzvaný dokument, ve kterém se nadále dají zakládat „studia“. Onshape nabízí mnoho zajímavých možností, běžného uživatele však zajímají tato tři hlavní studia - **Part Studio**, **Assembly** a **Drawing**.

Part Studio

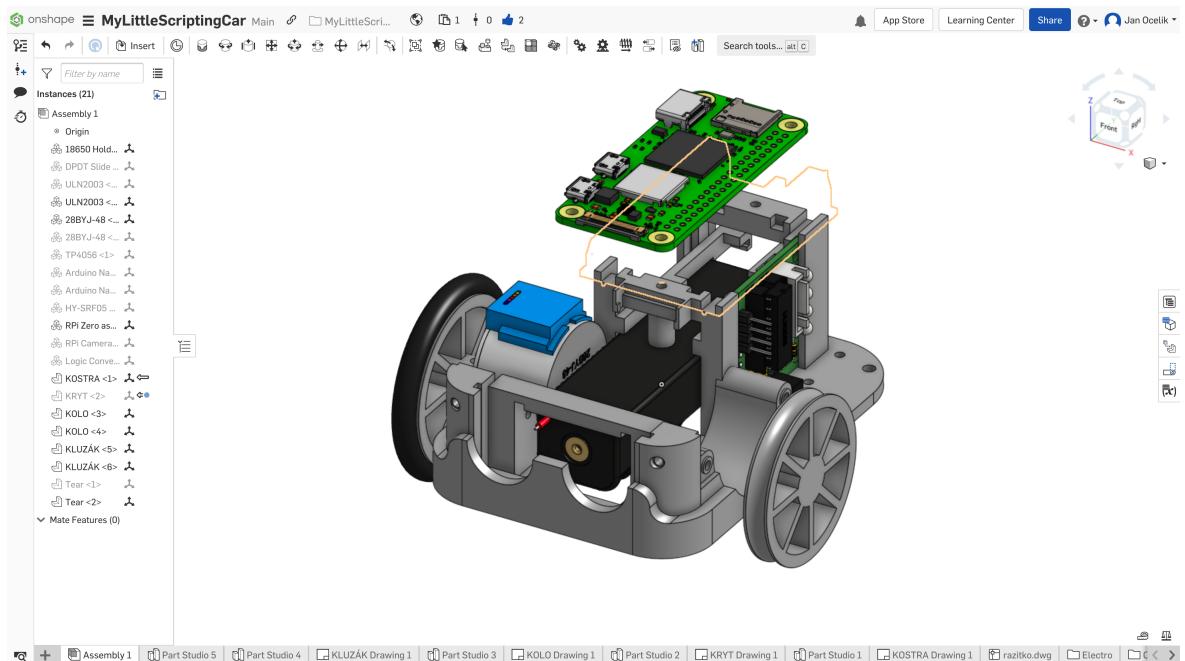
Part Studio slouží, jak už název napovídá, k vytváření jednotlivých dílů. Díly se vytvářejí a upravují pomocí příkazů, jako například **Sketch** (sloužící k sestrojení 2D náčrtu), **Extrude** („vytáhnutí“ náčrtu do třetího rozměru) nebo **Fillet** (zaoblení hran). Tyto příkazy lze vyvolat kliknutím na jejich ikonu v horní liště programu, přepsáním jejich názvu do **Search tools**, nebo, u některých více používaných, klávesovými zkratkami.



Obrázek 1: Part Studio

Assembly

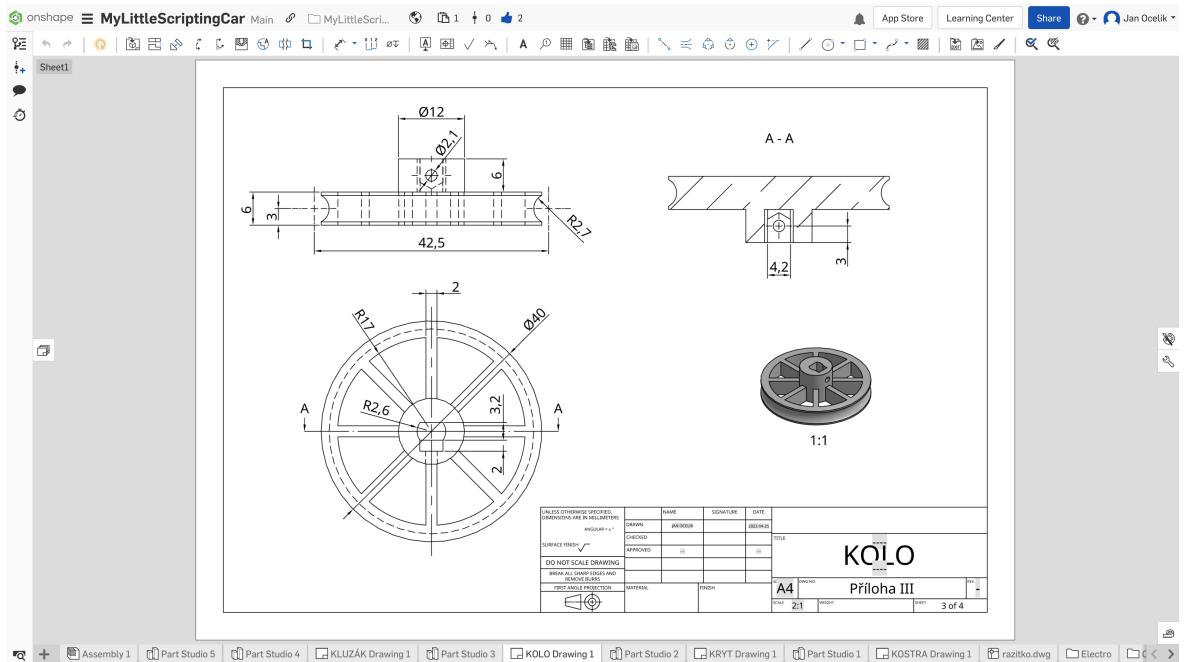
Další studio v programu Onshape, Assembly, slouží ke spojování (nebo alespoň aranžování) jednotlivých dílů. K tomuto účelu slouží příkazy, které se opět nechají vyvolat několika způsoby, včetně klepnutí na ikonu v horní liště studia. Další výhodnou funkcí tohoto studia je **Create Part Studio in context**, což dělá přesně to, co se v názvu píše; vytvoří studio dílu v kontextu se sestavou. Tato funkce může hodně usnadnit vytváření některých dílů.



Obrázek 2: Assembly

Drawing

Drawing se dá vyvolat jak samostatně, tak rovnou v kontextu s daným dílem a to jak z Part Studio, tak z Assembly. Uživateli stačí, když v seznamu dílů vybere danou položku a přes kontextové menu vybere možnost **Create Drawing of...**. Výkres se zde upravuje také pomocí příkazů, stejně jako v předešlých studiích. **Drawing** studio umožňuje také globální nastavení kót, písma, čar atd.



Obrázek 3: Drawing

1.2.2 Code OSS/VS Code

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "UNTITLED (WORKSPACE)".
 - src folder contains main.cpp, README.md, and EightStepMotor.h.
 - MLSC_arduino folder contains main.cpp, README.md, and EightStepMotor.h.
 - MyLittleScriptingCar folder contains .gitignore, platformio.ini, and README.md.
- Editor Area (Top):** Displays the content of main.cpp.

```
1 //include <Arduino>
2 //include <EightStepMotor.h>
3
4 #define PINCOUNT 4
5 #define STEPS 8
6 #define REVOLUTION 4096
7
8 int IPm = 15;
9 int timer_us = 60*1000L*1000L/REVOLUTION/rpm;
10 int pins[PINCOUNT] = {8, 9, 10, 11};
11
12 unsigned long last;
13
14 EightStepMotor Motor(REVOLUTION, pins[0], pins[1], pins[2], pins[3]);
15
16 void setup() {
17     // put your setup code here, to run once:
18     for (int i=0; i<PINCOUNT; i++) {
19         pinMode(pins[i], OUTPUT);
20     }
21     last = millis();
22     Serial.begin(9600);
23     Motor.setRPM(rpm);
24     Motor.setSteps(SL*REVOLUTION);
25     Serial.println(Motor.getSteps());
26 }
27
28 void loop() {
29     // put your main code here, to run repeatedly:
30     Motor.tick();
31     if (!Motor.getSteps()) { Motor.setRPM(0); } // This protects the motor from overheating
32 }
```
- Bottom Status Bar:** Shows the current file as "main.cpp - Untitled (Workspace) - Code - OSS". It also includes file navigation icons and a status message "Ln 12, Col 20 Spaces: 2 UTF-8 LF C++ PlatformIO".

Obrázek 4: Code OSS

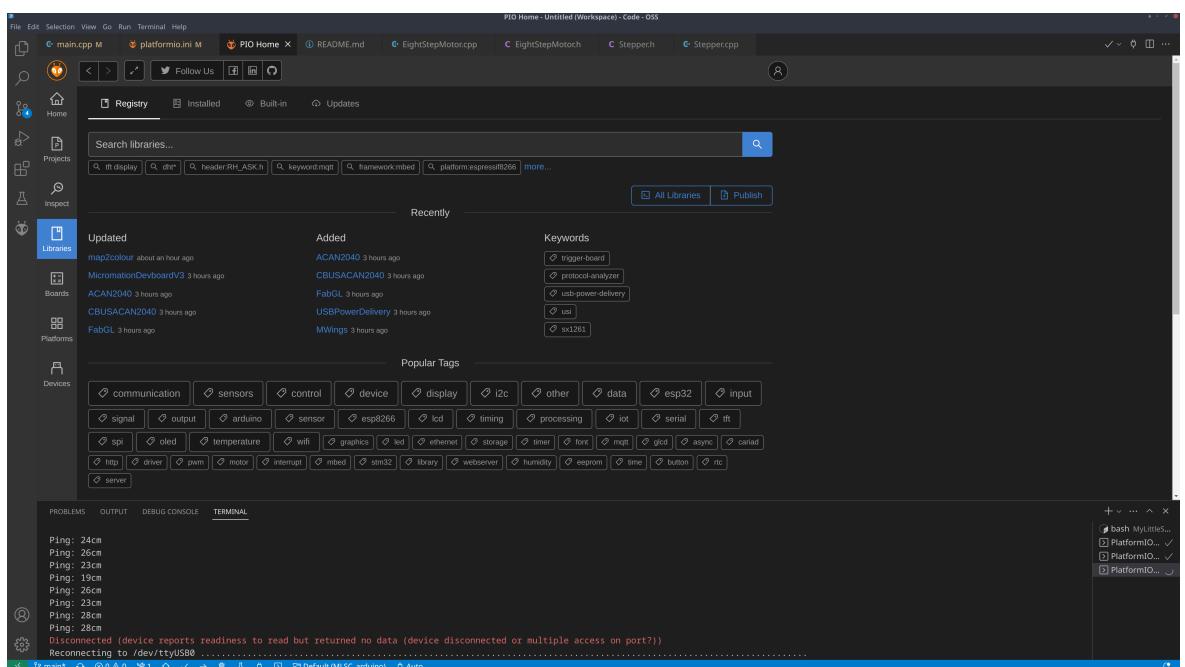
Code OSS je opensource IDE vyvíjené firmou Microsoft. Vzniklo jako konkurent programu Atom, který se nyní již nevyvíjí. Stal se oblíbeným hlavně díky podpoře obrovského

množství programovacích i značkovacích jazyků (např. python, c++, java, xml, html). Má také velice intuitivní ovládání a šikovně zvolenou paletu barev.

Code OSS a Visual Studio Code jsou téměř identické programy, jediným pozorovatelným rozdílem je absence některých rozšíření v „marketplace“ v Code OSS (i to se dá ale snadno opravit).

PlatformIO

Platformio je rozšíření do VS Code umožňující překlad C++ kódu a následný upload do předdefinovaných jednočipů (např. atmega328). Je to šikovný kus softwaru s několika užitečnými funkcemi navíc, jako např. automatická detekce zařízení, seznam dostupných knihoven, snadná instalace a integrace daných knihoven do projeků a další.



Obrázek 5: PlatformIO

1.3 Výrobní technologie

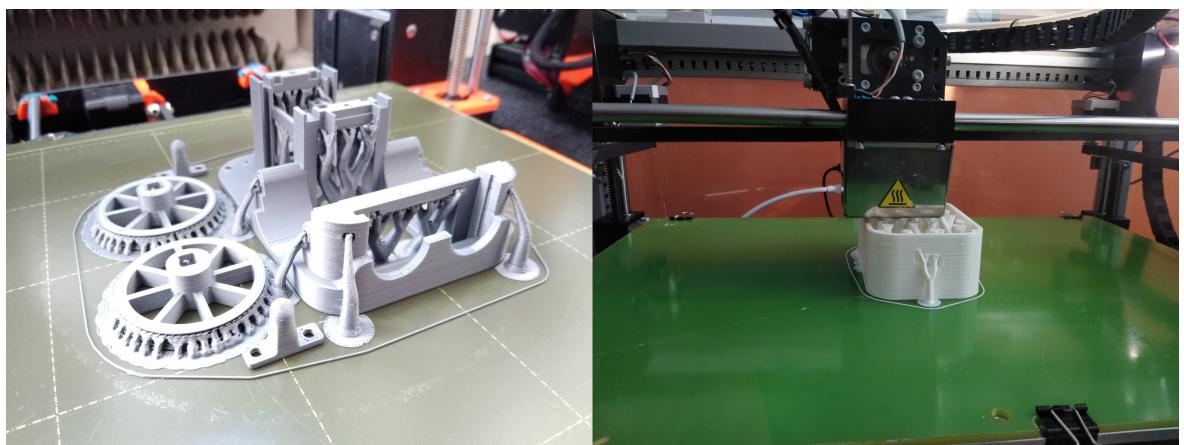
1.3.1 3D tisk

3D tisk je moderní technologie, která umožňuje vytvářet fyzické objekty ze 3D digitálních modelů. Tento proces začíná vytvořením digitálního modelu pomocí počítačového softwaru. Poté je tento digitální model převeden do formátu, který je kompatibilní s 3D tiskárnou. V průběhu tiskového procesu se materiál, obvykle termoplastického typu, zahřívá a aplikuje na specifická místa v souladu s digitálním modelem. Tento postup se opakuje vrstva po vrstvě, dokud není celý objekt hotov.

3D tisk má mnoho výhod, například umožňuje rychlé prototypování a výrobu malých sérií komplexních geometrických objektů. To se stává velmi užitečné v oblasti průmyslové výroby, kde může být vytvoření prototypu značně nákladné a časově náročné. Díky 3D tisku se náklady na výrobu mohou snížit a doba výroby může být zkrácena.

V posledních letech se cena 3D tiskáren snížila a uživatelé si mohou tisknout vlastní návrhy přímo z domova. Tato technologie také umožňuje vytvářet přizpůsobené a jedinečné předměty, což je velmi užitečné pro řadu různých aplikací, včetně medicíny, průmyslu a uměleckých oborů.

Díky jednoduchosti použití a možnosti tvorby téměř jakýchkoli tvarů je ideální volbou k výrobě kostry programovatelného auta.



Obrázek 6: Vytištěný model kostry, kol a kluzáků auta (vlevo) – Tisk modelu krytu (vpravo)

2 Popis konstrukce a vybrané moduly

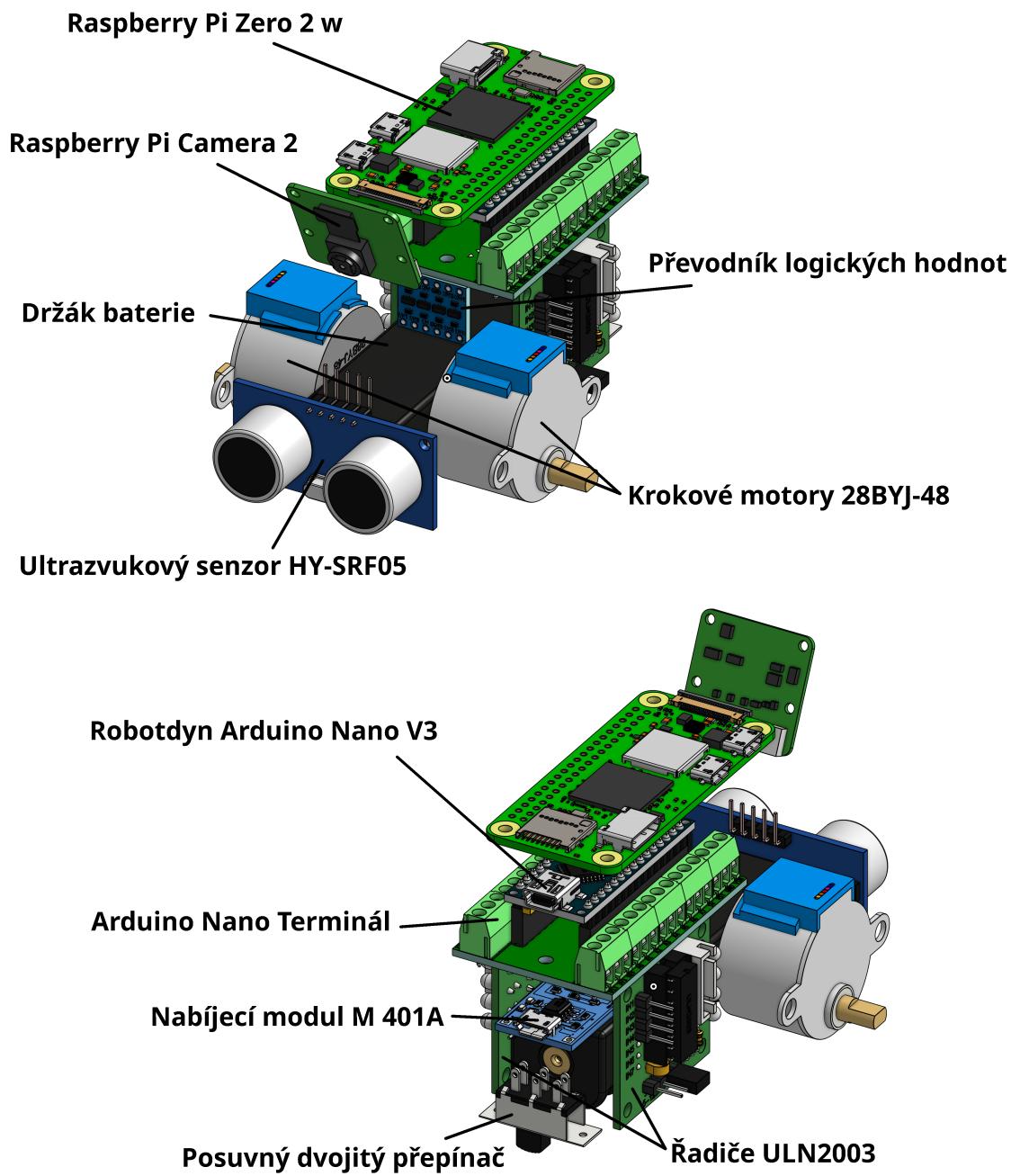
2.1 Vybrané moduly

Pro ovládání motorů bylo nutné vybrat „řídící jednotku“. Pro tento účel se nejen kvůli rozměrům jevila jako jasná volba Arduino Nano, ale kvůli ceně bylo lepší sáhnout po dostupnějším klonu: **Robotdyn Arduino Nano V3**. Piny na Arduinu jsou přístupné pouze ze spodní strany desky, což je vzhledem k jeho umístění v robotovi dosti nešikovné. Elegantním řešením problému je takzvaný terminál, který piny zpřístupní z boků.

Kvůli předpokládanému univerzálnímu využití robota bylo nutné zvolit také vhodný pohon. Tím se ukázaly být krokové motory. K motorům jsou potřeba také řadiče a to kvůli velkým proudům, které mohou motory procházet. Aby mohl být robot alespoň částečně autonomní, je téměř nutností přidat nějaké vstupní periferie, které poslouží k získávání zpětné vazby k řízení, jako například: **Ultrazvukový měřič vzdálenosti HY-SRF05**.

Pro správné fungování robota bylo nutné zvolit vhodné napájení. U autonomního zařízení je jasnou volbou nabíjecí baterie a díky své popularitě a rozšíření je nejlepší možností **Lithion baterie 18650** v kombinaci s dobíjecím modulem **M 401A** a step-up měničem **M 407**. Na baterii je také nutné vybrat vhodný držák a vzhledem k jejímu umístění je potřeba zvolit takový, který baterii udrží ve správné pozici i při hrubším zacházení s robotem. Je také nutné mít možnost robota nějak vypnout/odpojit (přerušit obvod), v ideálním případě napájecí obvod kompletně izolovat. Proto je dobré zvolit dvoucestný přepínač, například **posuvný dvojitý přepínač**.

Raspberry Pi Zero 2 w je vhodným modulem pro rozšíření funkcí robota, který se propojí s Arduinem (řídícím modulem) pomocí asynchronní sériové linky a umožní tak rychlejší zpracování údajů získaných z dalších rozšiřujících modulů, například z **Raspberry Pi Camera 2 NoIR**. Protože Raspberry a Arduino pracují na různých hladinách napětí, je potřeba pro bezpečné propojení a komunikaci mezi těmito moduly použít **převodník logických hodnot**.



Obrázek 7: Umístění modulů v robotovi

2.2 Konstrukce

Cílem této práce bylo navrhnut kostru programovatelného auta tak, aby bylo co nejkompaktnější a zároveň aby se zachovala snadná přístupnost ke všem integrovaným modulům.

Kostra je navržena tak, aby většina modulů mohla být nainstalována pouhým zasunutím do drážek a pomocí gravitace, zacvaknutím, nebo překrytím jiným modulem. Díky tomu bylo použito co nejméně šroubových spojů.

Krokové motory bylo třeba pevně ukotvit usazením do výrezů, které kopíruje tvar jejich těl, a zajištěním samořeznými šrouby.

Ultrazvukový senzor je zasunut do drážky, která jej těsně obklopuje ze tří stran. Zamezí se tím nežádoucím vibracím senzoru při pohybu robota.

Držák baterie je kvůli usnadnění výměny baterie umístěn na spodní straně tak, aby baterie byla přístupná bez nutnosti odstraňovat kryt robota. Napájecí obvod, včetně nabíjecího modulu, je snadno přístupný a může tak být v případě potřeby bez problémů vymontován.

Vypínač je umístěn na spodní straně vzadu a ke kostře pevně uchycen samořeznými šrouby, kvůli předpokládanému častému používání.

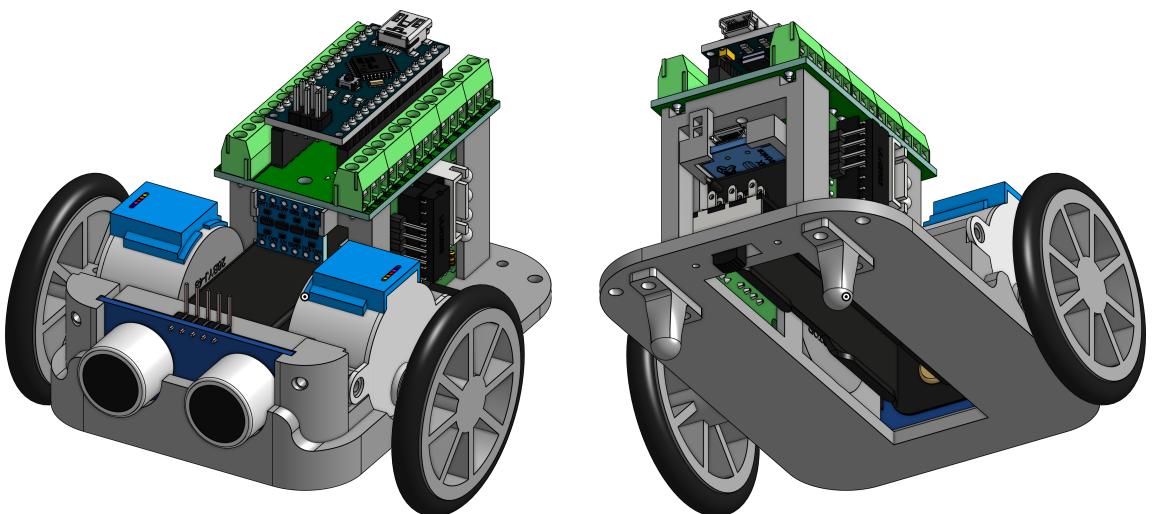
Arduino je umístěno tak, aby byl snadno přístupný jeho programovací konektor. Pomocí terminálu, do nějž je Arduino zasunuto, byly zpřístupněny všechny jeho piny (svorky). Samotný terminál je ukotven ke kostře samořeznými šrouby.

Vzhledem k vysoké kompaktnosti robota byly použity pouze dva krokové motory, tedy navržené auto má pouze dvě kola. Pro zajištění stability auta je nutné mít alespoň tři opěrné body (zde dvě kola a jeden kluzák). Kvůli umístění vypínače uprostřed robota bylo nutné zvýšit počet dodatečných opěrných bodů (kluzáků) na dva a umístit je po stranách vypínače.

Aby se zabránilo prokluzování kol na osách krokových motorů, jsou navržená kola vybavena integrovanou matičkou a šroubkem M2, které pomocí tření zabrání sklouznutí kola z osy.

Pro zvýšení přilnavosti kol k podkladu při pohybu robota byla použita toroidová těsnění (průměr 42mm/5,5mm) jako pneumatiky, která jsou usazena v drážce na kolech.

Základ robota obsahuje pouze Arduino a moduly, které zajišťují základní funkce: pohyb, napájení a sběr dat ze vstupní periferie (ultrazvukového senzoru).



Obrázek 8: Umístění modulů na kostře auta

Na zádech robota je provizorně umístěna kamera a Raspberry Pi, které slouží v této práci pouze k demonstraci řízení robota posíláním příkazů přes sériovou linku Arduinu a pomocí dat získaných z ultrazvukového senzoru a kamery připojené přímo do Raspberry Pi. Tímto propojením se, obrazně řečeno, z Arduina stává „mícha“, která předává příkazy z „mozku“ (Raspberry Pi) do periferií.



Obrázek 9: Zakrytované auto s rozšiřujícími moduly (Raspberry Pi)

3 Software

Hlavní činností Arduina je nízkoúrovňová správa základního hardwarového vybavení robota, tedy krokových motorů a ultrazvukového senzoru.

Pro řízení krokových motorů existuje několik knihoven, avšak žádná z nich nesplňovala požadavky na paralelizaci s ostatními procesy. Z tohoto důvodu jsem vytvořil vlastní knihovnu využívající možnosti paralelního ovládání motorů na základě zadané rychlosti a počtu kroků. Využil jsem objektově orientovaného přístupu a každý motor je tak nezávislou instancí společné třídy.

Podobný přístup využívá i opensource knihovna NewPing pro měření vzdálenosti od překážky ultrazvukovým senzorem.

Pro komunikaci mezi Arduinem a nedřízeným systémem, např. Raspberry Pi nebo PC, jsem si navrhl jednoduchý komunikační protokol umožňující snadné a intuitivní ovládání motorů a měření vzdálenosti od překážky (viz tabulka níže).

g získat hodnotu	r pravý motor l levý motor u ultrazvukový senzor	v rychlosť (rpm) n počet kroků	
s nastavit hodnotu	r pravý motor l levý motor u ultrazvukový senzor	v rychlosť (rpm) n počet kroků 0 zapnuto/vypnuto	±000 hodnota (-150 až +150) 0000 hodnota (0 až 9999)

Například:

"**srv+149**" – Arduino nastaví rychlosť pravého motoru na 14,9 rpm (otáček za minutu).

"**gu**" – Arduino vrátí hodnotu získanou z ultrazvukového senzoru.

V případě žádosti o získání hodnoty vrátí Arduino po sériové lince požadovanou hodnotu jen jako číslo.

4 Závěr

Téma této ročníkové práce jsem si zvolil na základě předchozích praktických zkušeností s návrhem vlastních robotů. Pro první konstrukce jsem používal na laserové gravírce vyřezávanou překližku a později mi rozšíření a zlevnění 3D tisku umožnilo používat i tyto moderní technologie. Kostru, kryt, kola a kluzáky tohoto robota jsem vytiskl na 3D tiskárnách na VOŠ a SPŠE Plzeň.

Arduino, ultrazvukový senzor, Raspberry Pi Zero 2 w a kamera jsou z mých vlastních zásob, ostatní moduly/součásti pro napájení a pohon jsem zakoupil přímo pro tento projekt na webu <https://techfun.sk>, <https://radiokus.cz> a <https://dratek.cz>.

Krokové motory by se mohli zdát jako nevhodně zvolené pro jejich pomalost, ale zvolil jsem je z důvodu jejich robustnosti a odolnosti, protože pro použití ve výuce není rychlosť rozhodující.

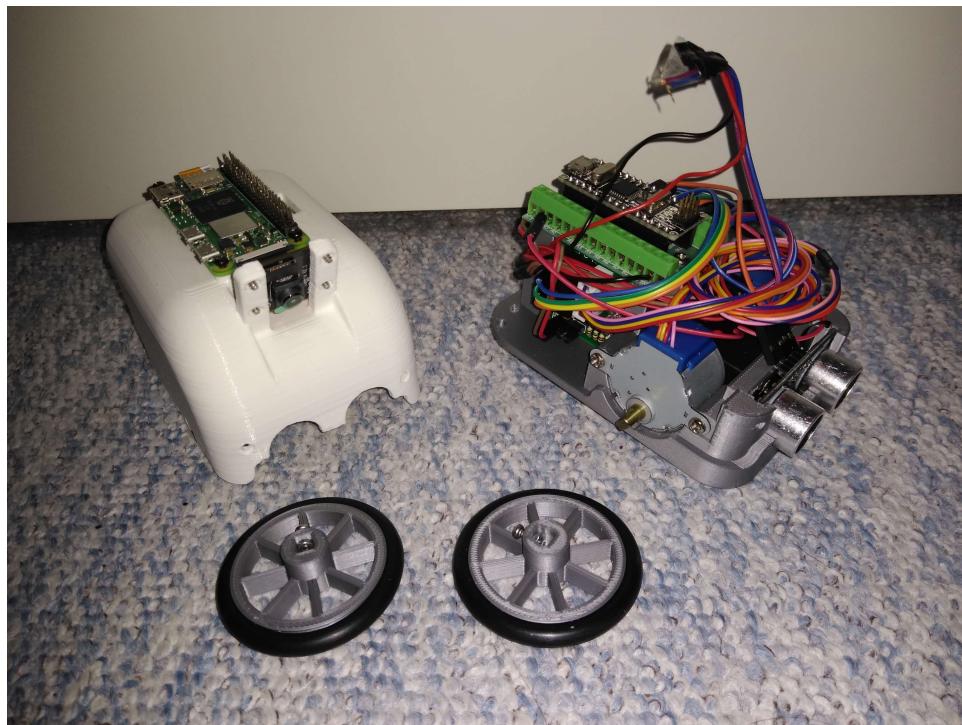
Pro krátkodobé použití s možností baterii kdykoli dobít je jedna baterie zcela postačující a robot tak může být kompaktnější.

Po praktických zkušenostech s návrhem této verze programovatelného auta bych do budoucna uvažoval o vlastním návrhu plošného spoje integrujícího funkce většiny použitých modulů. V tomto návrhu jsem také podcenil velikost potřebného prostoru pro vedení kabelů. Kryt je tedy velice těsný a při častější manipulaci hrozí poškození některých kabelů.

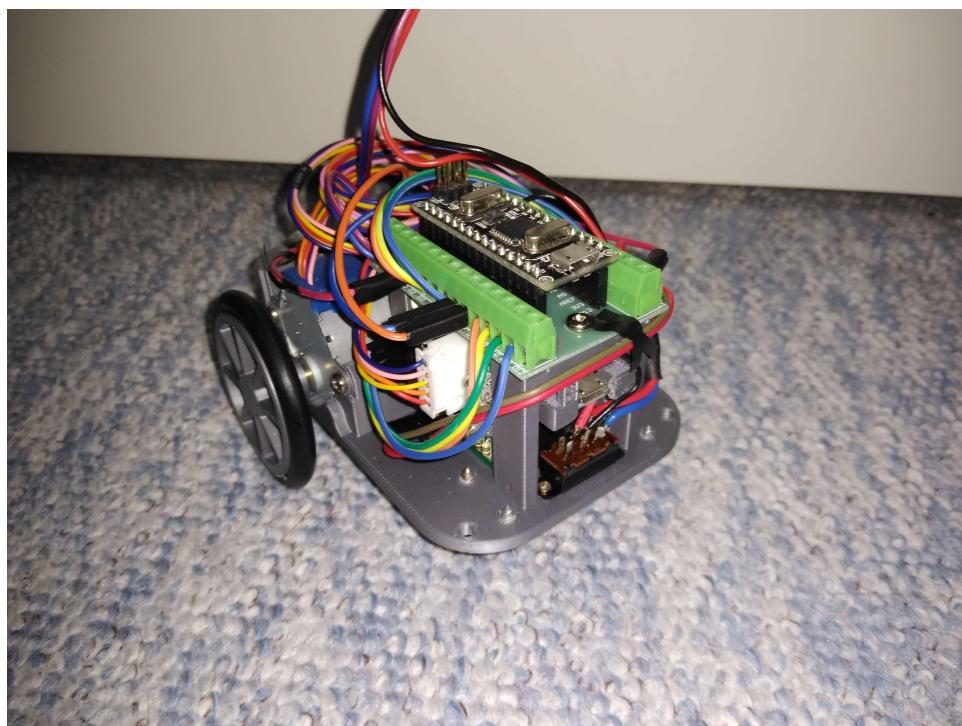
V této fázi Raspberry Pi s kamerou není plně integrován do robota a slouží pouze k demonstraci možností komunikace s robotem a v příštím návrhu bych jej chtěl pojmut jako rozšiřitelný kapotovaný modul pro větší odolnost.

Přínosem této ročníkové práce pro mě byly hlavně nové zkušenosti s návrhem šasi v <https://onshape.com> a tvorbou vlastní knihovny pro ovládání krokových motorů.

Také jsem si uvědomil, jak je důležité si práce na projektu rozvrhnout s dostatečným časovým předstihem a snažit se je dodržet.



Obrázek 10: Foto rozloženého auta



Obrázek 11: Foto kostry auta ze zadu



Obrázek 12: Foto složeného auta

Seznam obrázků

1	Part Studio	7
2	Assembly	8
3	Drawing	9
4	Code OSS	9
5	PlatformIO	10
6	Vytištěný model kostry, kol a kluzáků auta (vlevo) – Tisk modelu krytu (vpravo)	11
7	Umístění modulů v robotovi	13
8	Umístění modulů na kostře auta	15
9	Zakrytované auto s rozšiřujícími moduly (Raspberry Pi)	15
10	Foto rozloženého auta	18
11	Foto kostry auta ze zadu	18
12	Foto složeného auta	19

Seznam příloh

Příloha I.: KOSTRA - výkres

Příloha II.: KRYT - výkres

Příloha III.: KOLO - výkres

Příloha IV.: KLUZÁK - výkres

Příloha V.: Schéma zapojení - výkres

Příloha VI.: Kód projektu - MLSC_arduino.zip (elektronicky)

Příloha VII.: 28byj-48_uln2003_datasheet.pdf (elektronicky)

Příloha VIII.: BSS138.pdf (elektronicky)

Příloha IX.: camera-module-2-mechanical-drawing.pdf (elektronicky)

Příloha X.: camera-module-2-schematics.pdf (elektronicky)

Příloha XI.: Gravitech_Arduino_Nano3_0.pdf (elektronicky)

Příloha XII.: HY-SRF05-ETC.pdf (elektronicky)

Příloha XIII.: Logic_Level_Bidirectional.pdf (elektronicky)

Příloha XIV.: MT3608.pdf (elektronicky)

Příloha XV.: raspberry-pi-zero-2-w-productbrief.pdf (elektronicky)

Příloha XVI.: raspberry-pi-zero-2-w-reduced-schematics.pdf (elektronicky)

Příloha XVII.: TP4056-micro-usb.pdf (elektronicky)