



**Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Kotterovská 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	Švihla 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 uvadím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učteli 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
Úvod	6
1 Cíle a požadavky	7
1.1 Komponenty (moduly)	7
1.2 Šasi (tělo)	7
1.3 API	7
2 Návrhový software	8
2.1 Onshape	8
2.2 Code OSS/VS Code	10
3 Výrobní technologie	12
3.1 3D tisk	12
4 Vybrané komponenty (moduly)	13
4.1 Řídící modul - Arduino Nano klon	13
4.2 Terminál	13
4.3 Krokový motor a řadič	14
4.4 Přepínač	14
4.5 Ultrazvukový senzor	14
4.6 Převodník logických hodnot	14
4.7 Držák baterie	14
4.8 Nabíjecí modul	14
4.9 18650 Li-ion baterie	14
4.10 Raspberry Pi Zero 2	14

4.11 Raspberry Pi Camera 2 NoIR	14
---	----

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 byly 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 z praktického testování. 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. Tím může mít pozitivní vliv na jejich budoucí kariéru a na rozvoj robotiky v České republice.

1 Cíle a požadavky

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.

1.2 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 hardware za provozu.

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 nově příchozí uživatel zvládl bez jakékoli pomoci robota pomocí daného API/knihovny naprogramovat.

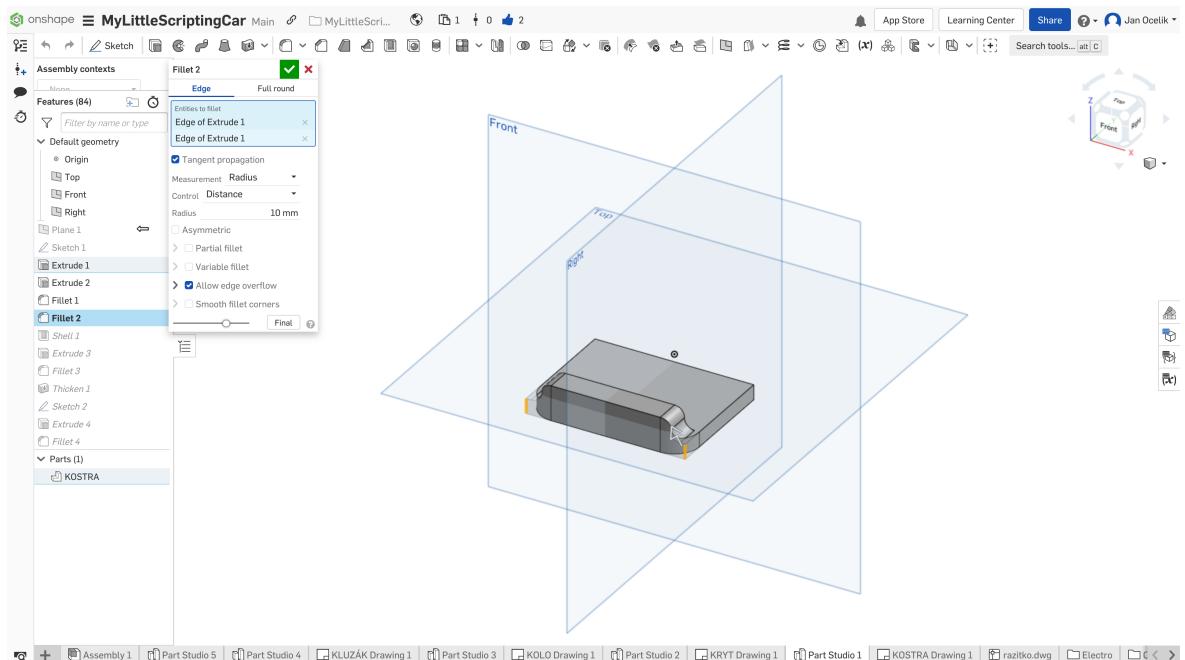
2 Návrhový software

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, typického uživatele však zajímají tato tři hlavní studia - **Part Studio**, **Assembly** a **Drawing**.

Part Studio

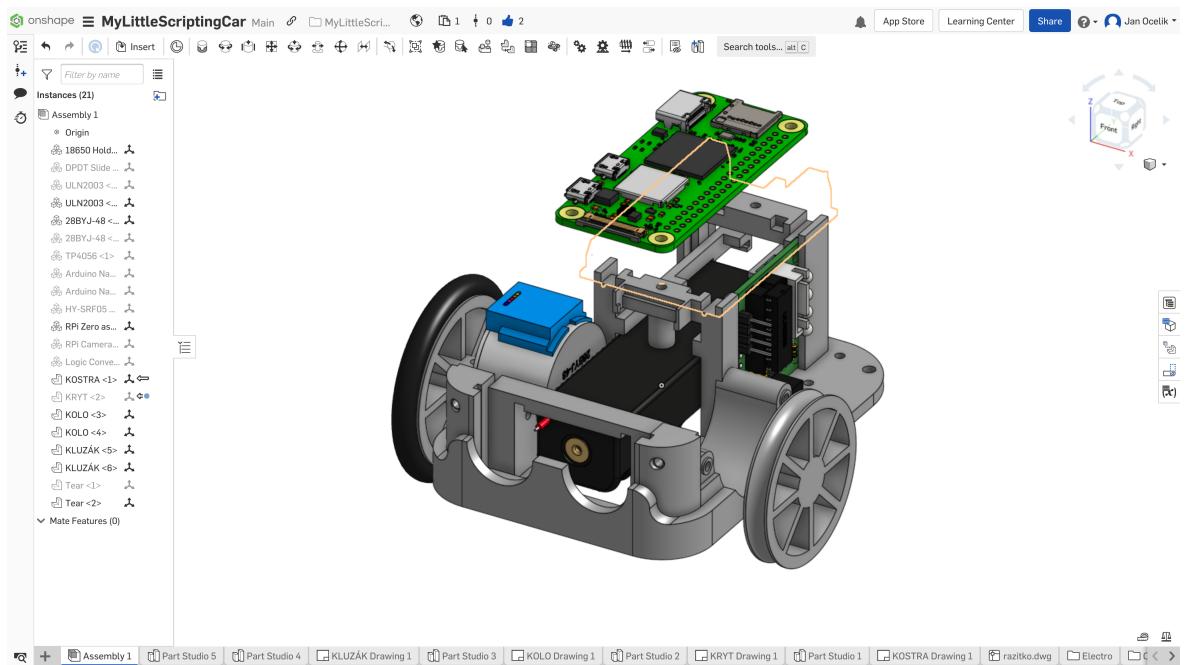
Part Studio slouží, jak už název napovídá, k vytváření jednotlivých dílů. Díly se vytvářejí a upravují pomocí příkazů, jako například **Sketch**, sloužící k sestrojení 2D náčrtu, **Extrude**, k „vytáhnutí“ náčrtu do třetího rozměru, nebo **Fillet**, k zaoblení hran. Tyto příkazy lze vyvolat kliknutím 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

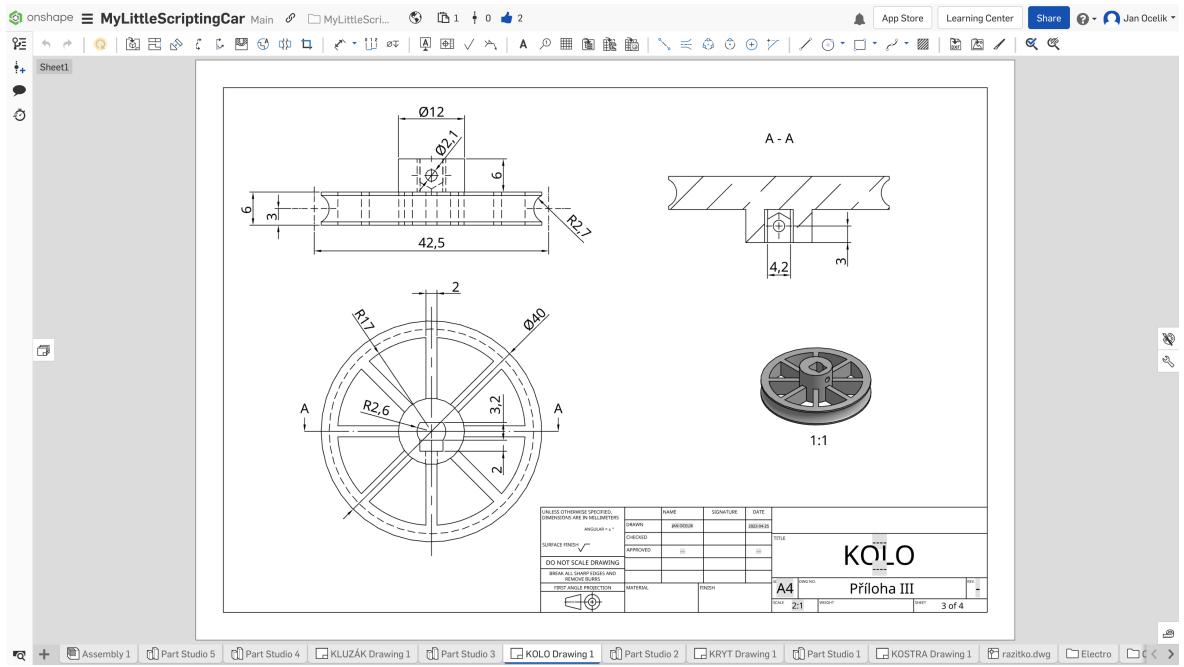
Další studio v programu Onshape, Assmebly, slouží ke spojování (nebo alespoň aranžování) jednotlivých dílů. K tomuto účelu slouží příkazy, které se, opět, nechají vyvolat několika způsoby, včetně klepnutím na ikonu v horní liště studia. Další výhodnou funkcí tohoto studia je **Create Part Studio in context**, 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, kdy 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

2.2 Code OSS/VS Code

```

File Edit Selection View Go Run Terminal Help
File Explorer View Editor Terminal
main.cpp x README.md EightStepMotor.cpp
MLSC_arduino > src > main.cpp > last
1 #include <Arduino.h>
2 #include <EightStepMotor.h>
3
4 #define PINCOUNT 4
5 #define STEPS 8
6 #define REVOLUTION 4096
7
8 int rpm = 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(8L*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 }
33

```

Obrázek 4: Code OSS

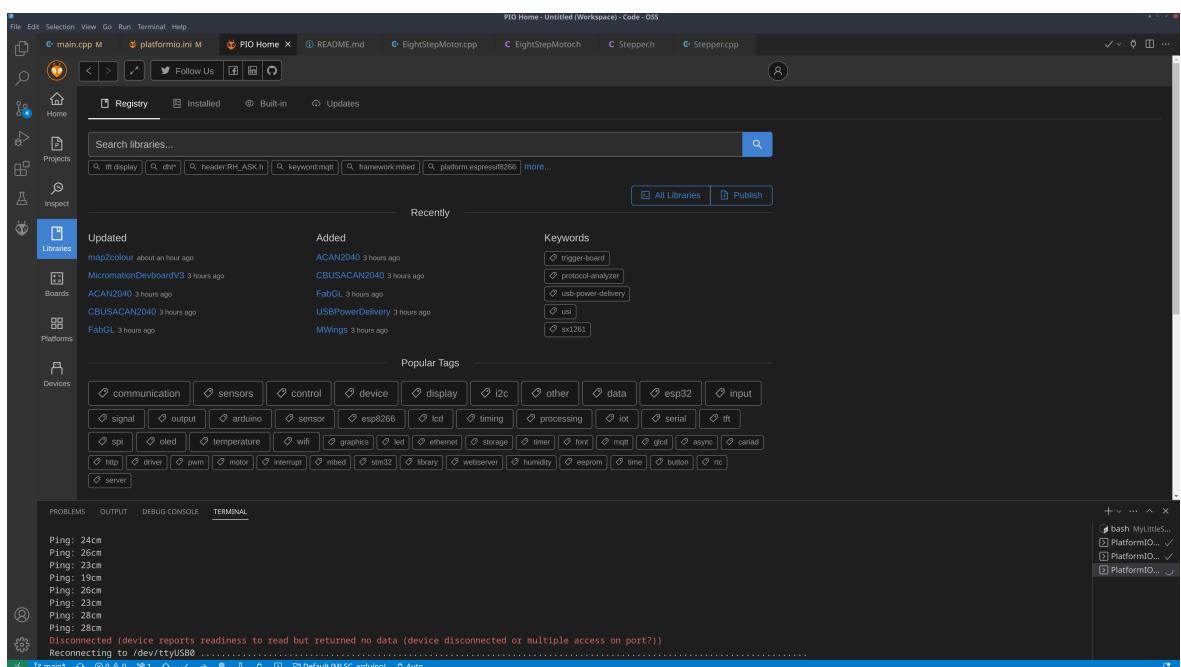
Code OSS je opensource IDE vyvýjené firmou Microsoft. Vzniklo jako konkurent programu Atom, který nedávno rozválcoval. Stal se tak oblíbeným hlavně díky podpoře obrov-

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

Code OSS a Vicual 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.



Obrázek 5: PlatformIO

3 Výrobní technologie

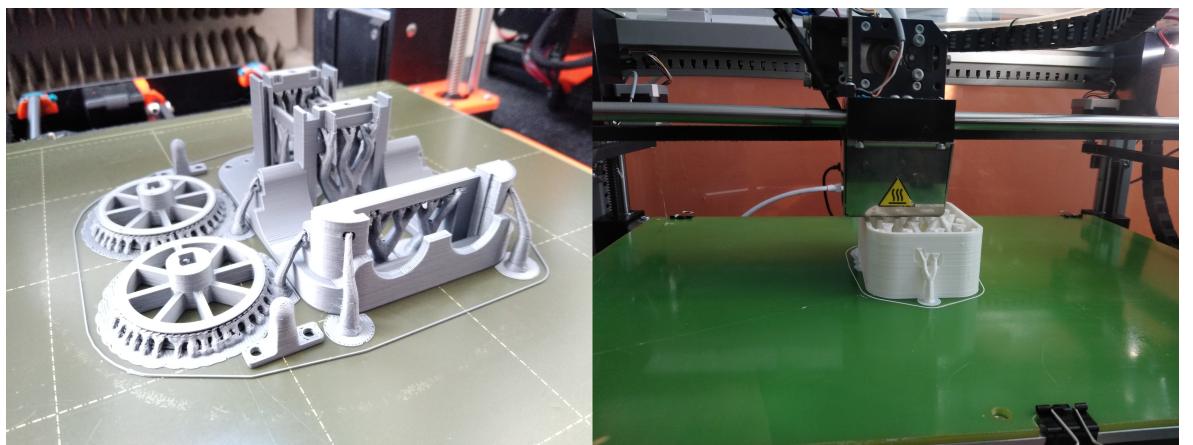
3.1 3D tisk

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.



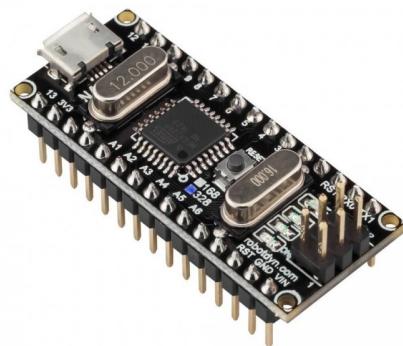
Obrázek 6: Vytisknutý model kostry, kol a kluzáků auta; Tisk modelu krytu

4 Vybrané komponenty (moduly)

4.1 Řídící modul - Arduino Nano klon

Pro ovládání motorů bylo nutné vybrat „řídící jednotku“. Pro tento účel a kvůli rozměrům se jeví jako jasná volba Arduino Nano, ale kvůli ceně bylo lepší sáhnout po dostupnějším klonu.

RobotDyn



Obrázek 7: Robotdyn Arduino Nano V3 [Zdroj: laskakit.cz]

4.2 Terminál

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.



Obrázek 8: Arduino Nano Terminál [Zdroj: techfun.sk]

4.3 Krokový motor a řadič

Kvůli předpokládanému univerzálnímu využití robota bylo nutné zvolit vhodný pohon. Tím se ukázaly být krokové motory. K motorům jsou potřeba také řadiče a to kvůli velkým proudům, které mohou motory procházet.



Obrázek 9: Krokový motor 28BYJ-48 a modul ULN2003 [Zdroj: dratek.cz]

4.4 Přepínač

4.5 Ultrazvukový senzor

4.6 Převodník logických hodnot

4.7 Držák baterie

4.8 Nabíjecí modul

4.9 18650 Li-ion baterie

4.10 Raspberry Pi Zero 2

4.11 Raspberry Pi Camera 2 NoIR