VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

VÝUKOVÝ TUTORIAL PRO POUŽITÍ ARDUINA V ROBOTICE

EDUCATIONAL TUTORIAL FOR USING OF ARDUINO IN ROBOTICS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN BERAN**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAROSLAV ROZMAN, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Beran Jan**
Program: Informační technologie
Název: **Výukový tutorial pro použití Arduina v robotice**
Educational Tutorial for Using of Arduino in Robotics
Kategorie: Vestavěné systémy

Zadání:

1. Nastudujte tutorialy týkající se Arduina a zaměřte se na řízení motorů a práci se senzory (enkodéry, sonary, IMU, IR senzory) a I2C sběrnicí.
2. Na základě nastudovaných tutorialů navrhněte vlastní tutorial, kde vysvětlíte princip Arduina, jeho programování, připojování senzorů a jejich programování. Dále vysvětlete možnosti propojení Arduina a dalších miniPC (Bluetooh, wi-fi a další). Navrhněte tutorial tak, aby se dal přizpůsobit různé délce výuky.
3. Podle navrženého tutorialu vytvořte slajdy (v ČJ a AJ) a jednotlivé programy pro Arduino. Ty musí být ve dvou verzích, kompletní a určené pro doplnění kódu.
4. Vytvořený tutorial otestujte na skupině uživatelů. Podle jejich připomínek proveďte případné změny.

Literatura:

- Tutorialy pro práci s Arduinem.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 31. října 2019

Abstrakt

Cílem této práce je vytvoření výukového tutoriálu pro využití Arduina v robotice za použití robota Trilobot. Tutoriál si klade za cíl být zaměřený na cílovou skupinu studentů ZŠ a výše se zájmem o robotiku a mikrokontrolery, ovšem bez pokročilých znalostí jejich fungování. V teoretické části budou shrnuty současné tutoriály pro platformu Arduino a podobné, spolu s návrhem přestavby robota Trilobot a návrhem tutoriálu. V praktické části bude realizováno jak vytvoření samotného robota, tak i příprava slajdů jak v českém, tak anglickém jazyce a zdrojových kódů k tutoriálu a provedení testování.

Abstract

The aim of this thesis is to create an educational tutorial for using Arduino in robotics with the Trilobot robot. The tutorial aims to be focused on the target group of elementary school students and above with an interest in robotics and MCUs, but without any advanced knowledge of their operation. The theoretical part will summarize the current tutorials available for the Arduino and similar, together with the design of the reconstruction of the Trilobot robot and the design of the tutorial. In the practical part, both the creation of the robot itself and the preparation of slides in both Czech and English languages and source codes for the tutorial and testing will be implemented.

Klíčová slova

Arduino, Trilobot, tutoriál, robotika, mikrokontroléry

Keywords

Arduino, Trilobot, tutorial, robotics, microcontrollers

Citace

BERAN, Jan. *Výukový tutorial pro použití Arduina v robotice.* Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Výukový tutorial pro použití Arduina v robotice

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana, PhD. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Beran
28. května 2020

Poděkování

Tímto děkuji vedoucímu mé práce, Ing. Jaroslavu Rozmanovi, PhD., za ponechání volnosti při realizaci této práce a průběžnou konzultaci.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Dnešní doba nahrává robotice, elektronice a programování. Ať již máme na mysli Průmysl 4.0, IoT – Internet věcí, Smart Home nebo třeba 3D tisk, všechno to jsou oblasti, které v poslední době nabývají na důležitosti.

Tato digitalizace společnosti klade i nové nároky jak na stávající pracovníky, tak na ty budoucí – nynější žáky a studenty, u kterých se alespoň elementární znalost programování jeví jako nesporná výhoda v budoucí kariéře. Například i v českém školství můžeme vidět snahy o rozšíření výuky programování. Spolu s tím se zvyšuje i množství různých produktů, které mají za cíl pomoci žákům a studentům proniknout do této jistě velmi perspektivní oblasti nejen ve škole, ale i v jejich volném čase. Roboti jsou pro tento cíl ideálním nástrojem, jelikož poskytuje studentům možnost vidět výsledky svého úsilí v reálném světě.

1.1 Cíl této práce

Cílem této práce je vytvoření výukového tutoriálu pro využití Arduina v robotice a zároveň i vytvoření samotného robota z původní platformy Trilobot.

Tento tutoriál by měl umožňovat využití a předvádění vytvořeného robota na různých akcích, pořádaných FIT VUT, jako například Den otevřených dveří nebo Letní škola (F)IT pro holky. Tutoriál ovšem může být využit i při běžné výuce, například jako doplňkový studijní materiál pro zpestření cvičení v předmětech zaměřených na mikrokontrolery. Díky plánované modularitě by měl být tutoriál přizpůsobitelný různé délce výuky, od malých ukázek, trvajících maximálně několik desítek minut, až po kompletní seznámení se vším, co Trilobot nabízí, které by zabralo řádově jednotky hodin. Zároveň je v plánu navrhnout a vytvořit tutoriál tak, aby reflektoval i různou úroveň znalostí žáků a studentů. Počítá se s úrovněmi tutoriálu pro žáky druhého stupně střední školy, studenty středních škol a studenty vysokých škol, případně ostatních nadšenců.

Jak již bylo zmíněno, Tutoriál bude spjat s robotem Trilobot, který pro tyto účely projde přestavbou. Tato přestavba bude zahrnovat montáž desky Arduino Mega, montáž všech potřebných senzorů a periferií, vytištění úchytů pro tyto senzory a periferie na 3D tiskárně a vytvoření programu pro kompletní ovládání Trilobota tak, aby ho bylo možné použít i mimo tento tutoriál.

Přínosem této práce by mělo být vytvoření jednoduché a uživatelsky přívětivé platformy, umožňující zájemcům o danou problematiku proniknout do programování robotů a mikrokontrolerů obecně. Oproti současným tutoriálům se bude méně zaměřovat na úplné základy a více se bude soustředit na práci se senzory a konkrétními komponenty. V neposlední řadě

se bude tutoriál snažit přimět studenty i k samostatnému vyhledávání informací (například v dokumentacích) a tím je navádět k postupům, které by používali i mimo tutoriál při vytváření vlastních projektů.

Kapitola 2

Současné tutoriály na téma Arduino

V této kapitole budou uvedeny některé z dostupných tutoriálů v českém a anglickém jazyce na téma „Arduino“. Jelikož neexistuje žádný objektivní způsob, jak jednoduše určit kvalitu tutoriálu, případně jinou metriku, podle které by se do tohoto výběru měly tutoriály zařazovat, byly vybrány mj. na základě předchozích zkušeností autora práce (Pojďme programovat elektroniku), na základě obecného povědomí o tutoriálech na toto téma (tutoriály na Arduino.cc, kniha Průvodce světem Arduina), případně podle kvality zdroje tutoriálu (tutoriály Massima Banziho, jednoho ze zakladatelů projektu Arduino).

V každém popisu zmíním mj. formu tutoriálu, jeho rozsah, oblast, které se tutoriál věnuje, předpokládanou úroveň znalostí čtenáře a krátce shrnu případné další aspekty stojící za zmínku, především ty, kterými se tutoriál liší od jiných. Pokud to bude vhodné, zmíním i některé konkrétní kapitoly tutoriálu, především pokud se budou věnovat oblastem jako enkodéry, sonary, IMU, IR senzory nebo I^2C sběrnici, případně dalším oblastem relevantním pro budoucí tutoriál. Na závěr uvedu krátký přehled nejdůležitějších vlastností každého tutoriálu.

2.1 Tutoriály v českém jazyce

V této části bude popsáno několik tutoriálů v českém jazyce, které je možné najít na internetu, případně zakoupit jako knihu. Jak již bylo řečeno, jelikož neexistuje žádný žebříček tutoriálů, podle kterého by bylo možné vybrat tutoriály do tohoto přehledu, zařazeny byly takové tutoriály, které pomáhaly i autorovi této práce v době, kdy se sám s platformou Arduino seznamoval.

Arduino tutoriál (ITnetwork.cz)

Tento tutoriál [19] z roku 2016 se nachází na jedné z největších českých webových stránek zaměřených na tutoriály o programování, ITnetwork [42]. Tutoriál je rozdělen do 15 kapitol, kde je každá kapitola uveřejněna jako samostatný článek. V prvních kapitolách je čtenář seznámen s Arduinem a programovacím jazykem Wiring, v dalších se poté probírá konkrétní problematika jako například práce s čidlem teploty nebo použití sběrnice I^2C .
Jelikož je tutoriál určen začátečníkům, popisy senzorů a problematiky obecně jsou často neúplné, sdělovány jsou pouze informace nutné pro zprovoznění daného senzoru. V kódu jsou

využité české názvy proměnných a funkcí, což někomu dává větší jistotu, jelikož je pro něj kód méně „cizí“, pro někoho ovšem může být kombinace češtiny a angličtiny nevyhovující. Tutoriál se v sedmé kapitole [20] zabývá měřením vzdálenosti pomocí sonaru HC-SR04. Pro co největší zjednodušení používá knihovnu NewPing [14], později i standardní způsob, popsaný v dokumentaci k sonaru [6]. Princip měření ovšem vysvětluje velmi stroze.
V jedné z dalších kapitol [21] je také popsána praktická stránka práce s I^2C sběrnicí, opět pouze na úrovni základní práce, tedy zapojení a jeden konkrétní příklad pro práci s LCD. Ten se bohužel od běžné práce s I^2C poněkud liší a uživatel může být zmaten při svém dalším setkání s touto sběrnicí.
Shrnutí:

- Forma a rozsah: 15 článků dostupných online
- Předpokládaná úroveň čtenáře: začátečník
- Specializace tutoriálu: Žádná, probírají se zde obecné základy a představují se konkrétní senzory.

Průvodce světem Arduina

I když je většina tutoriálů publikována pomocí internetu, tým HW Kitchen [15] vydal vlastní knihu – Průvodce světem Arduina [41]. V ní autoři na 240 stranách shrnují vše potřebné k tomu, aby člověk neznalý Arduina byl schopen po přečtení knihy s Arduinem sám dále pracovat a případně si získávat další informace.
Kromě samotné práce s Arduinem se čtenář seznámí i s jazykem Wiring (3.3), a to vždy prostřednictvím praktických ukázek. Kromě toho je v knize i několik projektů, na kterých si může čtenář vyzkoušet své schopnosti.
Oproti předchozímu tutoriálu je zde místo i na základní teorii, například teoretický popis řízení maticového displeje a podobně. Podrobnější recenzi prvního vydání této knihy sepsal mj. Martin Malý [29].
Kniha bohužel neobsahuje podrobnější informace o připojování senzorů, řízení motorů nebo I^2C sběrnici.

- Forma a rozsah: kniha, 240 stran
- Předpokládaná úroveň čtenáře: začátečník
- Specializace tutoriálu: Kompletní představení Arduina a způsobu jeho programování. Je zaměřený hlavně na obecný přehled a na to, aby byl čtenář dále schopen pracovat a získávat informace samostatně.

Tutoriály Arduino.cz

Tato česká webová stránka přímo o Arduinu je opět spjata s týmem okolo HW Kitchen [15], takže například Zbyšek Voda, hlavní autor Průvodce světem Arduina [41], přispívá i na tuto webovou stránku. Tutoriálům se zde věnuje celá sekce, obsahující několik samostatných tutoriálů. Jeden z nich – právě od Zbyška Vody – je vlastně základem výše zmíněné knihy. Kromě českých tutoriálů se zde nachází i tutoriál od Massima Banziho, spoluzakladatele Arduina, o kterém bude řeč v sekci 2.2.
Za zmínku stojí také sekce „Arduino projekty“ [16], kam přispívají dobrovolníci s, jak již

plyne z názvu, hotovými Arduino projekty. I když tato sekce není tutoriálem a není určena pro naprosté začátečníky, je skvělá pro načerpání inspirace ve chvílích, kdy uživatel Arduina hledá nápady nebo stávající řešení pro svůj nový projekt.

- Forma a rozsah: několik sekcí s různými tutoriály, případně ukázkami projektů.
- Předpokládaná úroveň čtenáře: začátečník až pokročilý
- Specializace tutoriálu: Různá podle konkrétního tutoriálu, každý tutoriál přistupuje k představení Arduina různě. Tutoriál Zbyška Vody například popisuje problematiku relativně podrobně, ale stále velmi přístupně. Tutoriál Arduino v příkladech popisuje vše, jak název napovídá, na příkladech.

Pojďme programovat elektroniku

I když se nejedná o tutoriál v pravém smyslu slova, přesto stojí tato série článků za zmínku. Jakub Čížek, redaktor Živě.cz [33], jednoho z velmi známých portálů o počítačích a IT obecně, zde již od roku 2016 pravidelně uveřejňuje články nejen o Arduinu, ale obecně o programovatelné elektronice [43]. Články jsou vcelku rozsáhlé jak svou délkou, tak zaměřením – v prvních článcích byly probírané stejné záležitosti jako v předchozích tutoriálech – tedy základy práce s Arduinem – ale v dalších článcích se autor zmiňuje i o nestandardních komponentách, například o e-ink displeji [46]), některé články věnuje i teorii, jejíž vysvětlování je sice omezené na minimum, ale poskytuje kompletní přehled. Vše je vysvětleno s praktickým přístupem a ukázkami.
Jeden z článků je mj. zaměřen i na měření vzdálenosti sonarem HC-SR04 [45], kde je vysvětleno jak použití senzoru, tak základní fyzika, která se za senzorem skrývá. V dalším článku [44] autor popisuje i práci s IR čidly.

- Forma a rozsah: Několik desítek článků na různá témata.
- Předpokládaná úroveň čtenáře: od začátečníka výše.
- Specializace tutoriálu: Žádná, zabývá se opravdu širokou škálou aspektů práce s programovatelnou elektronikou obecně. Nechybí ani tolik potřebná teorie, takže čtenář je poté schopen nejen slepě „pospojovat dráty“, ale minimálně tuší, co se děje na pozadí.

Arduino návody

I když se opět nejedná o klasický tutoriál, ale o sérii návodů, přesto je zde uvedena. Důvodem je, že na rozdíl od většiny ostatních tutoriálů obsahuje podrobnější informace o velkém počtu konkrétních modulů a senzorů, mimo jiné i těch, které jsou použity u robota Trilobot[1].
Tato webová stránka obsahuje zhruba 200 popisů různých senzorů a komponent, které je možné připojit k Arduinu a jemu podobným prototypovacím deskám [22].
Je zde shrnuta základní práce s enkodéry [24], IMU [31] i sonary [28] [23]. Autor již předpokládá základní znalost práce s Arduinem – naprogramování, připojení senzoru apod. Shrnutí:

[1] Jmenovitě senzory HC-SR04, Sharp GP2Y0A41SK0F a LCD displej připojený přes I^2C .

- Forma a rozsah: zhruba 200 článků.
- Předpokládaná úroveň čtenáře: pokročilý (ovládá základní práci s Arduinem).
- Specializace tutoriálu: Praktická práce s konkrétními senzory. Vzhledem k tomu, že návody jsou součástí e-shopu s komponentami pro Arduino a slouží částečně i k propagaci obchodu (na konci článku jsou odkazy na použité komponenty), autor se je snažil psát kvalitně a srozumitelně, což se mu podařilo.

Hradla, volty, jednočipy: Úvod do bastlení

Tato kniha [30] poskytuje čtenářům kompletní přehled o domácí elektrotechnice, která se nejlépe popisuje slovem „bastlení“. V knize jsou zahrnuty informace od těch nejzákladnějších (fyzikální povaha elektřiny a elektrických součástek), přes využití výše zmíněných prvků, až právě po Arduino, které je zde prostředkem praktické realizace obvodů a systémů, které se v knize popisovaly. Velkou výhodou této knihy je, že kromě základních informací o Arduinu (které zde nejsou hlavním tématem) dává k dispozici i ucelený soubor informací o celém odvětví elektrotechniky.

- Forma a rozsah: kniha o zhruba 500 stranách.
- Předpokládaná úroveň čtenáře: začátečník, je ovšem vyžadována znalost fyziky na úrovni zhruba prvního ročníku SŠ.
- Specializace tutoriálu: elektrotechnika obecně, Arduino je pouze jedno z témat.

2.2 Tutoriály v anglickém jazyce

Jelikož valná většina IT světa mluví anglicky, dá se v angličtině najít mnoho kvalitních Arduino tutoriálů. Níže bude krátce popsáno několik z nich.

Arduino Starter Kit – Video Tutorials by Massimo Banzi

Massimo Banzi, spoluzakladatel projektu Arduino, vytvořil pro svůj projekt mimo jiné tutoriál [8] a knihu *Getting started with Arduino* [9]. Tutoriál tvoří 10 krátkých videí na serveru YouTube, kde autor vysvětluje základní principy fungování Arduina na malých „projektech“. Kromě jiného vysvětluje i jak pomocí Arduina a seriové linky kreslit na obrazovku hostitelského počítače.
Autor se soustředí spíše na základy, proto zde nejsou probírány senzory typu sonar, IMU a podobně.

- Forma a rozsah: 10 tematických videí na serveru YouTube.
- Předpokládaná úroveň čtenáře: začátečník.
- Specializace tutoriálu: V tutoriálech autor detailně popisuje kód tak, aby mu skutečně každý porozuměl, opět ale sledující odstiňuje od technických detailů, které se schovávají v pozadí a nejsou nezbytně nutné pro fungování projektů.

Tutoriály na webu Arduino.cc

Arduino.cc [1] je oficiální internetovou stránkou projektu Arduino a jako taková má i rozsáhlou tutoriálovou sekci. Kromě jiného jsou zde umístěny i vestavěné příklady, které si může každý spustit z Arduino IDE a nahrát do své desky. Na webu jim ovšem nechybí ani krátký textový doprovod a ilustrace znázorňující zapojení komponent jako například LED diody, tlačítka a podobně.
Podsekce vyhrazená přímo tutoriálům je opět spíše skladem mnoha desítek projektů, které vytvořili lidé z celého světa. Každý má proto trochu jiné pojetí i úroveň a opět jsou určeny spíše pro pokročilejší uživatele, hledající inspiraci a nové nápady.
Na rozdíl od většiny předchozích tutoriálů jsou zde k nalezení dvě podsekce s informacemi o samotném hardware a software Arduina a potřebné teoretické informace. Protože Arduino je open-hardware, dají se zde nalézt i informace, jak si postavit vlastní verzi Arduina ze základních komponent [17].
Kromě výše uvedeného návodu se zde nacházejí mj. i informace o mapování pinů z čipů ATmega na piny na Arduinu, informace o programátoru, princip překladu a podobně.
Jelikož je základna uživatelů a přispěvatelů na stránky arduino.cc opravdu rozsáhlá, dají se zde nalézt články na mnoho různých témat, například o enkodérech [38], sonarech [40] nebo I^2C sběrnici a knihovně `Wire.h` [7]

- Forma a rozsah: různé sekce se stovkami článků, řazenými do tematických oblastí.
- Předpokládaná úroveň čtenáře: od začátečníka přes pokročilého až po zkušeného uživatele.
- Specializace tutoriálu: jelikož se nejedná jen o jeden tutoriál, je oblast, kterou celá sekce pokrývá, skutečně velmi rozsáhlá.

2.3 Shrnutí a hlavní rysy současných tutoriálů

Tutoriály jsou vesměs různorodé, ale u většiny z nich se dají najít některé společné charakteristiky. První takovou charakteristikou je například všeobecné zaměření tutoriálů, kdy se málokterý z nich věnuje nějaké konkrétní oblasti použití. Většina z nich se být co nejvíce obecná, což bohužel u některých, zejména kratších, tutoriálů způsobuje, že tutoriál nic neprobírá na takové úrovni, aby to případnému čtenáři bylo užitečné. Dalším poměrně dobře sledovatelným společným znakem je zaměření na začátečníky, kdy se pravděpodobně počítá s tím, že pokročilejší uživatelé již nebudou tutoriály vyžadovat a budou schopní získávat potřebné informace i odjinud.

Co se týče formy tutoriálů, většina z nich je psaná jako série samostatných článků dostupných on-line, občas doplněných videi. Hlavní výhodou tohoto způsobu publikace je poměrně snadné vyhledávání a členění tutoriálu, kdy student nemusí složitě hledat jednotlivé části v obsahu. Většinou na sebe navazují pouze volně a lze je tedy používat i samostatně, ale není to pravidlem.

Samostatným typem (nebo spíše doplňkem) tutoriálů jsou pak výše zmíněné knihy: Průvodce světem Arduina [41] a Hradla, volty, jednočipy [30]. Obě tyto knihy poskytují oproti většině tutoriálů širší přehled o Arduinu a jeho programování (v případě druhé zmíněné knihy dokonce o celém oboru elektrotechniky). Daní za velké množství kvalitních informací

je ovšem délka, která může začátečníka poněkud vyděsit. V neposlední řadě jsou obě knihy v papírové podobě placené, i když se dají zdarma získat jako e-knihy[2].

U některých tutoriálů se vyskytují i určité společné problémy. Jedním z nich, patrným především u těch tutoriálů, které jsou zaměřené na úplné začátečníky, je upozaďování teorie a předávání jen nezbytně nutných informací o dané problematice. Tím se na jednu stranu zkracuje čas tvorby funkčního celku, na druhou stranu student často není schopný znalosti přenést na jiný příklad. Tento jev je k vidění například na tutoriálu [42], kde autor často jen stroze popisuje předložený kód (byť i ten je okomentovaný). Celkovou slabinou tutoriálů je možná až přílišné zjednodušování, kdy je sice čtenář schopen relativně rychle Arduino zapojit a naprogramovat dle zadání, ale plně nerozumí zapojení nebo kódu. To jim později ztěžuje další práci s Arduinem, kdy si musí vším projít znovu a musí zjišťovat podrobnosti a dodatečné znalosti odjinud.

V souvislosti s předchozím bodem také vyvstává otázka, jestli je student po absolvování tutoriálu schopen nějakého samostatného posunu, tedy zda je schopný nabyté vědomosti používat i v jiných situacích a případně si je rozšiřovat.

Některé tutoriály toto zohledňují a poskytují kompletní přehledy informací nad rámec právě řešeného problému. Toto je vidět především u dvou výše zmíněných tištěných publikací nebo u tutoriálu „Pojďme programovat elektroniku“. Především u tutoriálů pro začátečníky se ale podobné otázky moc neřeší a po absolvování tutoriálu je tedy student odkázán na samostudium a hodiny hledání relevantních informací k již jednou probrané oblasti.

Z obecného pohledu je například patrné i to, že se žádný z tutoriálů nezaměřuje na konkrétní oblast, například na chytrou domácnost či přímo na robotiku. V této oblasti je tedy jistě prostor pro nové tutoriály, jelikož zde prakticky nemají rovnocenné konkurenty.

[2]Hradla, volty, jednočipy jsou k dispozici na https://knihy.nic.cz/files/edice/hradla_volty_jednocipy.pdf a Průvodce světem Arduina na https://arduino.cz (ve druhém případě je třeba zadat svůj e-mail). Obě knihy jsou distribuované pod licencí Creative Commons a jejich elektronická podoba je tedy plně legální.

Kapitola 3

Platforma Arduino a robot Trilobot

Platforma Arduino byla zvolena z toho důvodu, že, jak je podrobněji vysvětleno dále, je primárně určená jako výuková platforma a přímo se počítá s tím, že by byla použita při studiu. Další nespornou výhodou je, že platforma má kolem sebe velmi silnou komunitu, která vytváří velké množství materiálů pro samostudium i tvůrčí činnost na této platformě: tutoriály, projekty a další.

Open-source a open-hardware povaha platformy přispívají k tomu, že existuje i velké množství klonů desek Arduino, které jsou finančně velmi dostupné, stejně jako moduly a senzory pro platformu Arduino, a proto si případní zájemci o tuto problematiku mohou pořídit svou vlastní vývojovou desku a vybavení k ní a pokračovat s experimentováním i v domácích podmínkách.

Důvod pro volbu konkrétní desky, Arduino Mega, spočívá v jejím velkém množství pinů a obecně větší ploše, není tedy nutné tolik přemýšlet o zapojování a případné rozšíření je jednodušší. Zároveň byla deska k dispozici a nemusely se tedy pořizovat nové vývojové prostředky.

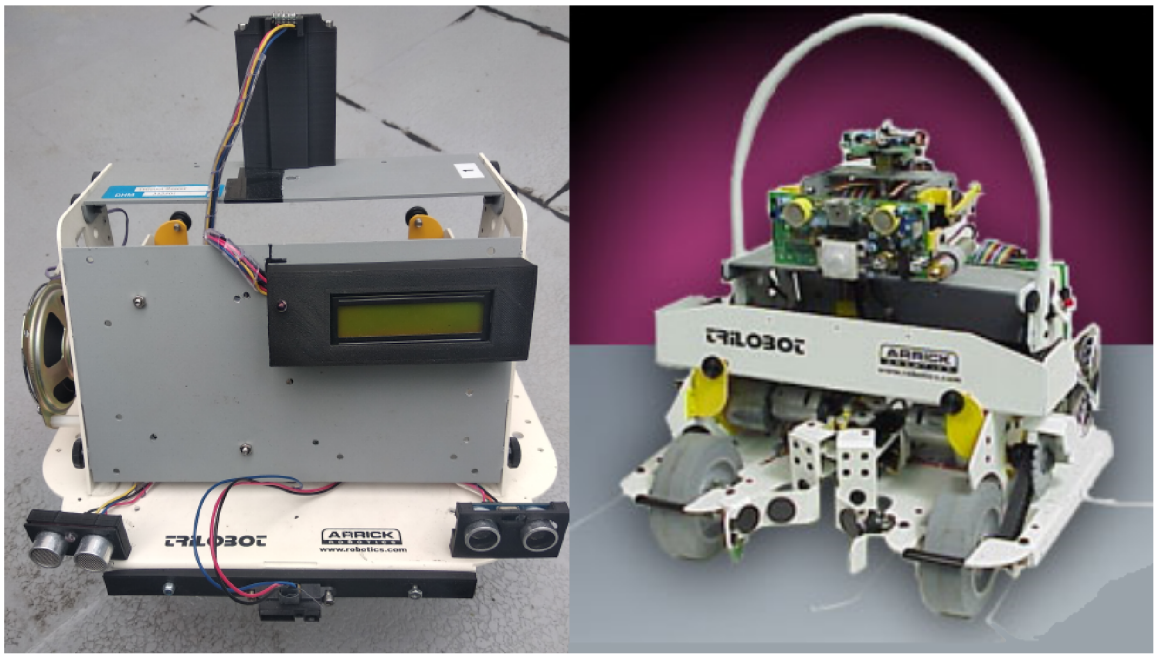
Trilobot je robotem určeným pro výukové a experimentální účely. Původně byl osazen čipem typu 8051. V současném stavu byl kus dostupný pro tuto práci bez mikrokontroleru a periferií, které byly dodány odděleně. Kromě kostry byl Trilobot osazen pouze motory Faulhaber 16002, koly a piezo reproduktorem.

Výhodou Trilobota je jeho kovové tělo, které by mělo vydržet i nešetrné zacházení, které může v případě začátečníka hrozit. Jeho nevýhodou je poněkud větší hmotnost.

3.1 Základní informace

Arduino je podle oficiálních stránek [1] elektronická platforma, která si klade za cíl být snadno použitelná a přístupná i těm, kteří nemají s programováním mikrokontrolerů žádné zkušenosti. První deska typu Arduino vznikla v roce 2005 s cílem pomoci studentům a těm, kteří nikdy předtím neprogramovali, například designérům a umělcům, zapojovat vlastní obvody a vytvářet vlastní elektronická zařízení, použitelná pro jejich potřeby.

Díky tomu, že je celý projekt Arduino open-source a open-hardware, vytvořila se kolem něj mohutná komunita, která se stará například o vytváření knihoven, tvorbu podpůrných materiálů (knih, dokumentace, tutoriálů) a podobně. Open-hardware povaha projektu také umožnila mnoha lidem vytvářet si vlastní klony Arduino desek. Toho využily hlavně čínské



Obrázek 3.1: Nová a původní podoba Trilobota. Obrázek vpravo přejat z [36].

e-shopy, které zaplavily trh mnoha levnými klony desek Arduino[1]. Díky tomu si každý může pořídit vlastní Arduino kompatibilní desku za nepříliš vysokou částku a experimentovat s ní v domácích podmínkách.

Celá platforma Arduino se skládá z několika součástí: samotných vývojových desek, jazyka Wiring a vývojového prostředí Arduino IDE. Níže budou ve stručnosti představeny všechny zmíněné části.

3.2 Přehled Arduino desek

Za zhruba 15 let existence platformy Arduino bylo vytvořeno mnoho různých desek, z nichž je ovšem zdaleka nejpoužívanější deska s názvem Arduino UNO. Kromě ní ovšem existuje i verze Mega (v současné době konkrétně Mega 2560), jejímž hlavním rozdílem je větší počet GPIO pinů a větší paměť, nebo desky Nano a Micro, kde byla hlavním kritériem velikost. Kromě těch ještě existují exotičtější desky, které se již tolik nepoužívají. Příkladem budiž například Yún s mnohem výkonnějším čipem, díky němuž může na desce fungovat i odlehčená linuxová distribuce, případně Lilypad, deska primárně zaměřená na tvorbu nositelné elektroniky.

Nejběžnější desky (UNO, Mega, Nano, Micro, ale i další) používají procesory od firmy Atmel, konkrétně typy ATmega328(P), ATmega32U4 nebo ATmega2560, všechny pracující na frekvenci 16 MHz. Paměť programu a dat je oddělená, je tedy použito Harvardské architektury počítače. Velikost pamětí se liší podle konkrétní desky.

Arduino UNO je první a dosud i zdaleka nejpoužívanější deskou platformy Arduino. V současné době se již prodává deska ve své třetí revizi (značeno „R3“, případně „Rev3“ za jménem). Deska používá procesor ATmega328. Její hlavní předností je kompromis mezi

[1] Více o čínských klonech a derivátech (odvozeninách, které nekopírují žádnou konkrétní desku) lze nalézt například zde [10]

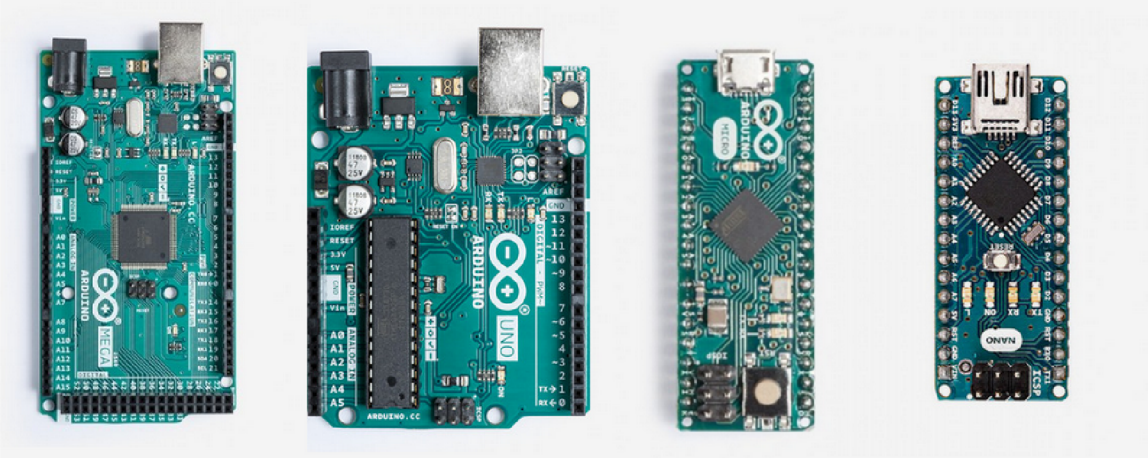
velikostí, snadným přístupem k pinům a cenou. Pro Arduino UNO je také koncipována většina tzv „shieldů“, rozšiřujících desek, které se nasadí na všechny piny a dodávají desce dodatečnou funkcionalitu, například internetovou konektivitu. V současné době jsou ovšem tyto shieldy na ústupu. Programování a napájení probíhá přes vestavěný USB-B port, přičemž napájení může být řešeno i pomocí vestavěného kulatého konektoru nebo přímo přes adekvátní piny Vcc a GND.

Arduino Mega je rozšířením desky Arduino UNO o větší paměť a větší počet pinů, což je také jeho hlavní výhodou. Nachází využití tam, kde již nedostačuje počet pinů Arduina UNO, případně je také vhodně pro začínající studenty, jelikož větší velikost desky umožňuje lepší zapojování kabelů.

Arduino Nano má opět prakticky totožnou funkcionalitu s deskou Arduino UNO, ale vydalo se opačným směrem než Arduino Mega; Nano je vytvořeno s cílem být co nejmenší při zachování kompatibility s nepájivými poli. Programování probíhá přes mini-USB oproti klasickému USB-B portu.

Menší rozměry předurčují Arduino Nano k vestavbě do finálních projektů, k použití v modelech letadel a všude tam, kde velikost a hmotnost hrají zásadní roli.

Arduino Micro je stejně jako Arduino Nano zástupcem miniaturních desek určených spíše pro trvalé zabudování do projektu, kde záleží na velikosti (IoT zařízení a podobně). Hlavním rozdílem je ovšem použitý procesor: ATmega32U4, který má zabudovaný USB převodník. Výhodou je především to, že Arduino Micro se může chovat jako standardní vstupní periferie počítače, tedy například jako myš nebo klávesnice a lze si takové zařízení naprogramovat. Zde ovšem může být problémem počet dostupných pinů.



Obrázek 3.2: Zleva desky Arduino Mega, UNO, Micro a Nano. Jednotlivé fotografie pochází z [2].

3.3 Arduino IDE a jazyk Wiring

Jak již bylo zmíněno, Arduino je názvem projektu a zároveň platformy, sdružující pod sebou jak hardwarové prostředky, tak vlastní jazyk a vývojové prostředí. V následující části bude stručně popsán jazyk Wiring, Arduino IDE a princip programování desek Arduino.

Jazyk Wiring

Na úvod je pro ujasnění terminologie třeba uvést, že samotný jazyk Wiring byl vytvořen v roce 2003 jako diplomová práce Hernanda Barragána [12]. Později, v roce 2005, Massimo Banzi[2] naklonoval repozitář jazyka Wiring a spolu se svým týmem ho začal vyvíjet pod platformou Arduino [11]. Samostatný jazyk Wiring sice stále existuje, ovšem je na okraji zájmu. Pojem „jazyk Wiring“, tedy v této práci (a v drtivé většině případů i jinde na internetu) odkazuje na klon původního jazyka Wiring pod platformou Arduino a v tomto smyslu zde také bude používán.

Jazyk Wiring měl za cíl při svém vytvoření zjednodušit práci umělcům a designerům, kteří používají elektroniku a kteří se do té doby museli učit relativně složité prototypovací jazyky, které v té době byly k dispozici. Základem tohoto jazyka je jazyk C++[3]. Označení „jazyk“ ve spojitosti s Wiringem ovšem může být diskutabilní, po technické stránce má spíše povahu rozsáhlé knihovny nebo frameworku pro jazyk C++.

Jeho základní filozofií je zapouzdřit složitou interakci s mikrokontrolery a nahradit je jednoduše použitelnými funkcemi. Například pro nastavení pinu jako výstupního není třeba nastavovat žádné bity v registrech, postačí zavolat funkci `pinMode()`, která vše provede a tím odstíní programátora od samotného mikrokontroleru. To také umožňuje velmi jednoduše programovat širokou škálu mikrokontrolerů pomocí stejných funkcí. V jazyce Wiring lze programovat kromě desek Arudino i například desky od společnosti Espressif s čipy ESP32 a ESP8266.

Jednotlivé programy pro Arduino se nazývají„sketche“ a používají vlastní příponu `.ino`. Tato přípona je přitom vyžadována pouze u hlavního souboru, který obsahuje funkce `setup()` a `loop()`. Pokud je tedy výsledný kód rozdělen do více souborů, ostatní soubory již používají standardní přípony jazyka C nebo C++ (`c., .cpp, h.`). Celý projekt přitom musí být umístěn ve složce, jejíž jméno se shoduje se jménem hlavního souboru.

Základní struktura programu pro Arduino v jazyce Wiring se dělí do dvou hlavních funkcí: `setup()` a `loop()`. Ve funkci `setup()` se nachází ta část kódu, kterou je nutné provést pouze jednou. Většinou se tedy jedná o nastavení pinů (zda-li budou dané piny použité jako vstupní nebo výstupní), inicializaci připojených komponent a podobně. Funkce `loop()`, jak již název napovídá, vykonává hlavní smyčku programu, tedy například odečet dat ze senzorů nebo odesílání dat, případně další komunikace s okolím.

3.4 Základy práce na platformě Arduino

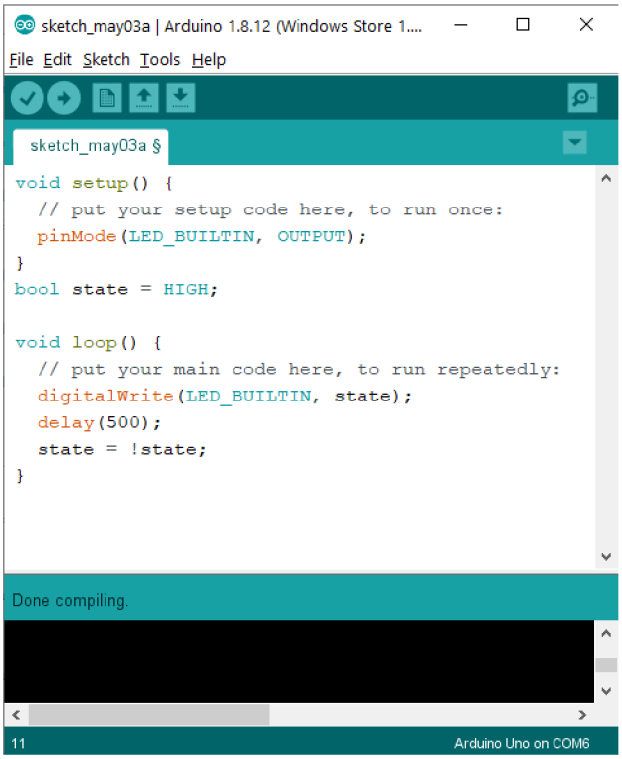
V této části bude krátce shrnuto a prakticky předvedeno, jak s Arduinem pracovat, jak do něj nahrávat programy, připojovat senzory a jak provádět další nezbytné úkony.

3.4.1 Arduino IDE

Jak již bylo zmíněno, oficiálním vývojovým prostředím pro platformu Arduino je Arduino IDE, velmi jednoduché vývojové prostředí, které je ve své podstatě jen textový editor s funkcí nahrávání programů a zvýrazňování syntaxe. Arduino IDE nepodporuje například našeptávání nebo ladicí funkce. Problémem je i práce ve více souborech, kdy sice mohou být otevřené zároveň, ale Arduino IDE nepodporuje rychlý přesun na místo definice/deklarace.

[2] Massimo Banzi byl dle Hernanda Barragána vedoucím jeho diplomové práce.

[3] Při psaní programů pro Arduino lze využívat všechny dostupné konstrukce tohoto jazyka.



Obrázek 3.3: Ukázka programu v jazyce Wiring v Arduino IDE.

Na obrázku 3.5 je vidět hlavní obrazovka vývojového prostředí Arduino IDE. Kromě hlavní části, do které se píše samotný kód (1), se ve spodní části nachází místo pro výpis ladicích informací při nahrávání (2). Vlevo nahoře pod standardní navigační lištou se nachází (zleva doprava) tlačítka pro překlad, nahrání programu, vytvoření nového souboru, otevření stávajícího souboru a uložení (3). O průběhu překladu a nahrávání je uživatel informován ve spodní části obrazovky.

Připojení desky Arduino a její nastavení se provádí nejprve fyzickým připojením a poté přes nabídku „Tools“, kde se vyberou správné položky v nabídkách „Board“, „Processor“ a „Port“. První dvě položky jsou dány deskou, která je aktuálně používaná. U originálních desek Arduino se COM port rozpozná jednoduše: v závorce za názvem portu je napsán typ desky, která je na daném portu připojená. U neoriginálních desek je třeba metody pokus-omyl, pokud je v nabídce více COM portů. Případně je možné vyvolat nabídku COM portů při připojené a při odpojené desce. Port, který při odpojení desky zmizí, je s největší pravděpodobností portem, na kterém je deska připojená. Nyní stačí desku opětovně připojit a vybrat identifikovaný COM port.

Následné programování probíhá tlačítkem „Upload“ v horní části obrazovky (na obrázku 3.5 v oblasti (3)). Pomocí nabídky „File->Open“ nebo klávesové zkratky „Ctrl + O“ je možné otevřít stávající soubor, přes „File->New“ nebo „ctrl + N“ vytvořit nový. Minimálně hlavní soubor (tedy ten s funkcemi `setup()` a `loop()`) musí mít příponu `.ino`, ostatní soubory již mohou mít standardní příponu zdrojových kódů v jazyce C, C++ nebo hlavičkových souborů (`.c, .cpp, .h`). Pod „File->Examples“ se také dají najít příklady použití Arduina, stejně tak jako příklady z přídavných knihoven, pokud nějaké jsou nainstalované.

Obrázek 3.4: Popis prostředí Arduino IDE.

3.4.2 Instalace přídavných knihoven

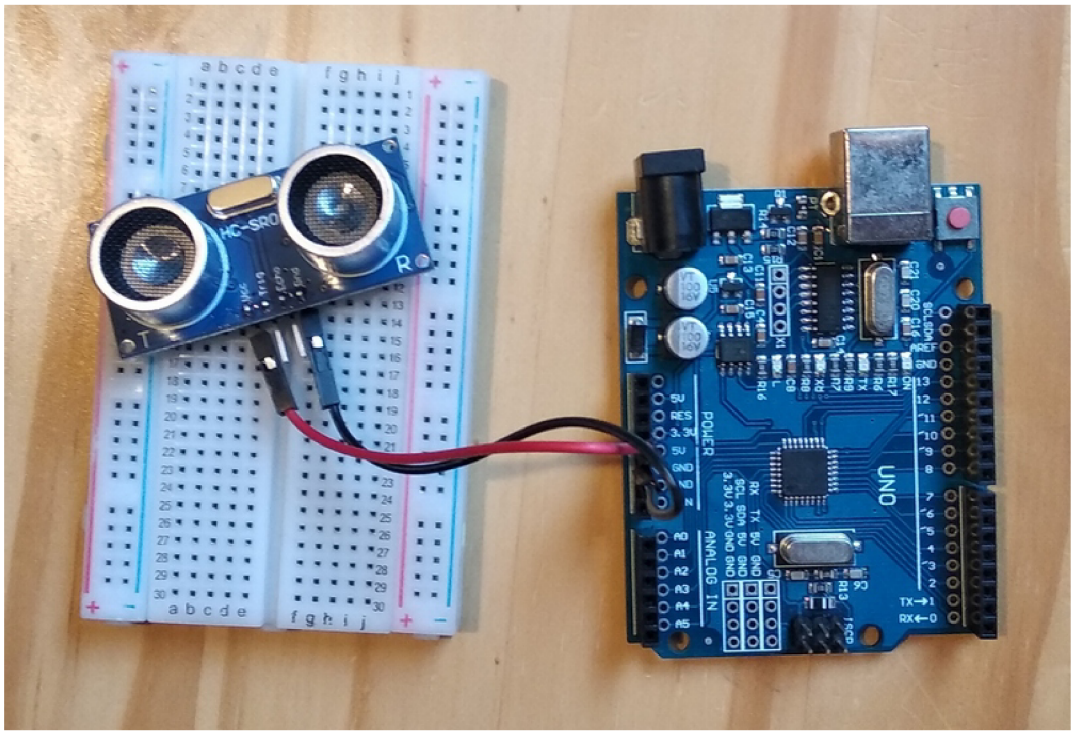
I když Arduino přichází s velkým počtem předinstalovaných knihoven, občas je nutné doinstalovat další. To se provádí přes nabídku „Sketch->Include library->Manage libraries“ v případě standardních Arduino knihoven. Pokud bude nutno nainstalovat knihovnu, která v této nabídce není (například ze serveru GitHub), provádí se tak přes „Sketch->Include library->Add .ZIP library“. Knihovnu je přitom nutné mít v .zip archivu. Po těchto krocích je doporučeno restartovat Arduino IDE.

3.4.3 Připojování senzorů

Připojování senzorů je realizováno prostřednictvím propojení pinů na desce Arduino a jejich protějšků na příslušném senzoru/periferii. Modely UNO a Mega mají piny vyvedené do pinových polí pro snazší připojování dupont kabelů, modely Nano a Micro slouží pro vestavění do konkrétního systému/projektu, proto mají piny vyvedené pouze do bodů připravených pro napájení kontaktů. Většina senzorů je stejně tak připravena na propojení přes univerzální dupont kabely a je tedy snadné je připojit[4]. Pro připojování často slouží i nepájivé

[4]Druhým častým typem konektorů jsou konektory typu JST. V závislosti na mezerách mezi piny mohou, ale nemusí být kompatibilní.

pole. Nepájivá pole se používají hlavně v situacích, kdy se realizuje obvod z jednotlivých základních součástek a bylo by nepřehledné je propojovat pouze pomocí kabelů.



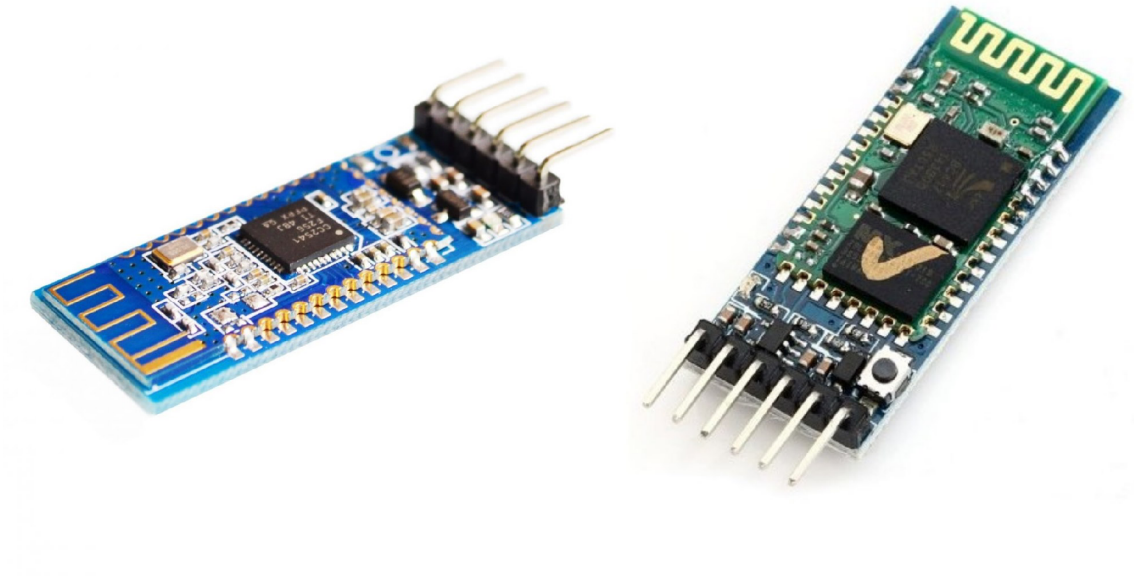
Obrázek 3.5: Ilustrace připojení senzoru HC-SR04 k desce Arduino UNO. Pro přehlednost je připojené pouze napájení, nikoli datové vodiče. V pozadí vidno nepájivé pole.

3.4.4 Komunikace mezi více deskami

Arduino samotné často slouží primárně pro prvotní seznámení se světem mikrokontrolerů, přesto je ale na něm díky jednoduchosti jeho programování postaveno mnoho větších či menších projektů. Jedním z mnoha příkladů mohou být vlastní chytré komponenty pro „Smart Home“, vytvořené na míru každému uživateli. Takové komponenty by jejich vývojáři rádi propojili mezi sebou – například v případě vlastnoručně postavené chytré domácnosti – nebo například s mobilním telefonem. Arduino samotné nemá žádnou WiFi, Bluetooth ani rádiovou konektivitu, ovšem pro všechny tři zmíněné způsoby komunikace existují moduly, které Arduinu komunikaci umožňují.

Bluetooth

Bluetooth je jednou z možností, jak mohou různé desky komunikovat navzájem mezi sebou, případně i s chytrým telefonem. Výhodou je poměrně nízká spotřeba a rozšířenost, nevýhodou poté poněkud omezený dosah, který záleží na použitém modulu. V případě Arduina se nejčastěji používají moduly HC-05 [27] a HM-10 [25]. První jmenovaný používá Bluetooth ve verzi 2.0 a má dosah řádově vyšší jednotky metrů. Druhý jmenovaný již používá Bluetooth 4.0 a zároveň podporuje i Bluetooth Low Energy (BLE), čímž se hodí zvláštně pro komponenty, které by měly fungovat na baterii. Jeho dosah se již pohybuje okolo několika desítek metrů v ideálních podmínkách.



Obrázek 3.6: Zleva: moduly HM-10 a HC-05 [25] [27]

WiFi

WiFi se v poslední době stala pokud ne nejrozšířenějším způsobem propojování komponent všeho druhu, tak minimálně tím nejvíce diskutovaným. Tato technologie má zřejmé výhody, například možnost přenášet relativně velké objemy dat vysokou rychlostí na velké vzdálenosti díky Internetu, možnost komunikace velkého počtu zařízení a mnoho dalších. Bohužel má tato technologie, zvláště při použití s mikrokontrolery, i řadu nevýhod. Tou nejvážnější je pravděpodobně otázka bezpečnosti. Jsou známy případy, kdy byly napadeny komponenty chytré domácnosti s internetovou konektivitou a útočníci získali přístup do celé domácí sítě [13].

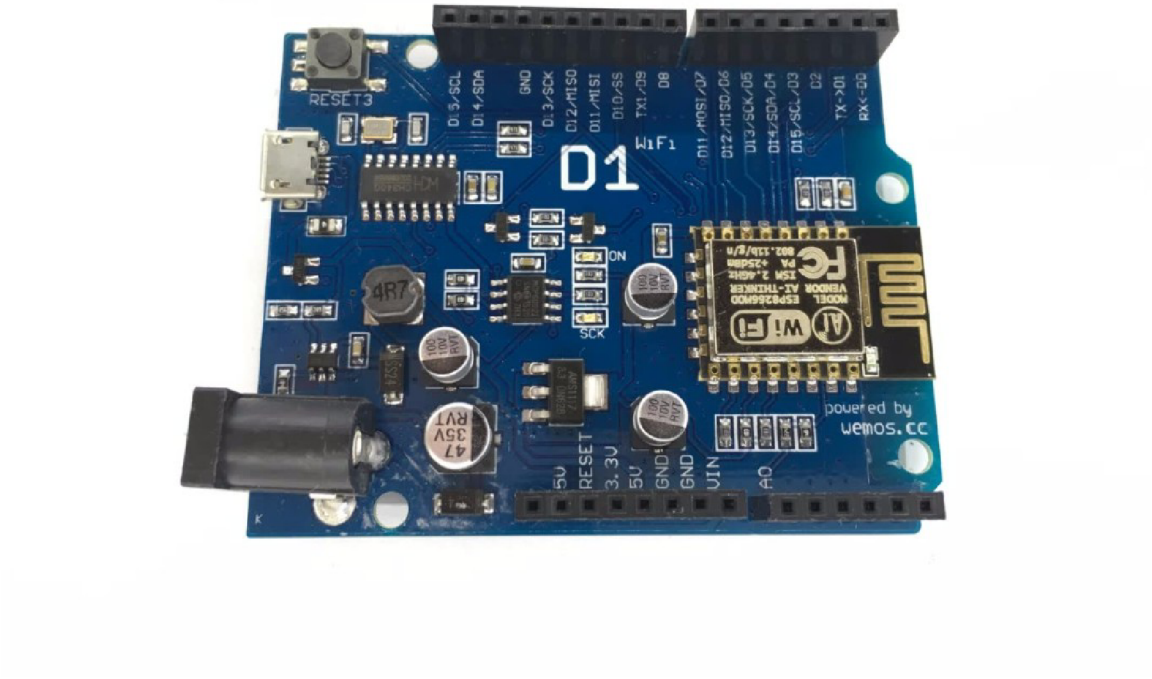
Použití WiFi spolu s Arduinem je v současné době komplikované a příliš se nepoužívá. Existoval rozšiřující WiFi shield pro Arduino UNO, ale v současné chvíli již není v prodeji a oficiálně není ani podporovaný.

Mnohem častěji se využívají vývojové desky, které jsou založené na čipech ESP32 a ESP8266, například Wemos D1 na obrázku 3.8. Velikost a rozložení pinů je téměř identické s deskou Arduino UNO. Největším rozdílem je pracovní napětí 3.3 V, na rozdíl od 5 V u Arduina UNO a Mega. tato deska ovšem nativně podporuje komunikaci přes WiFi v pásmu 2.4 GHz jak v módu klienta, tak v módu stanice.

Programování desek s čipem ESP32/ESP8266 je možné i přímo v Arduino IDE, stačí pouze přidat podporu těchto desek. Podrobný návod je dostupný například na [26].

Rádiová komunikace

Rádiová komunikace není příliš rozšířená, pravděpodobně kvůli malé podpoře ze strany počítačů a chytrých telefonů, ovšem umožňuje při velmi malé spotřebě dosáhnout velkého dosahu, který se může pohybovat až v řádech nízkých stovek metrů a je tedy ideální volbou pro případ, kdy musí Arduino komunikovat v místech, kde není WiFi signál a Bluetooth již svým dosahem nedostačuje.

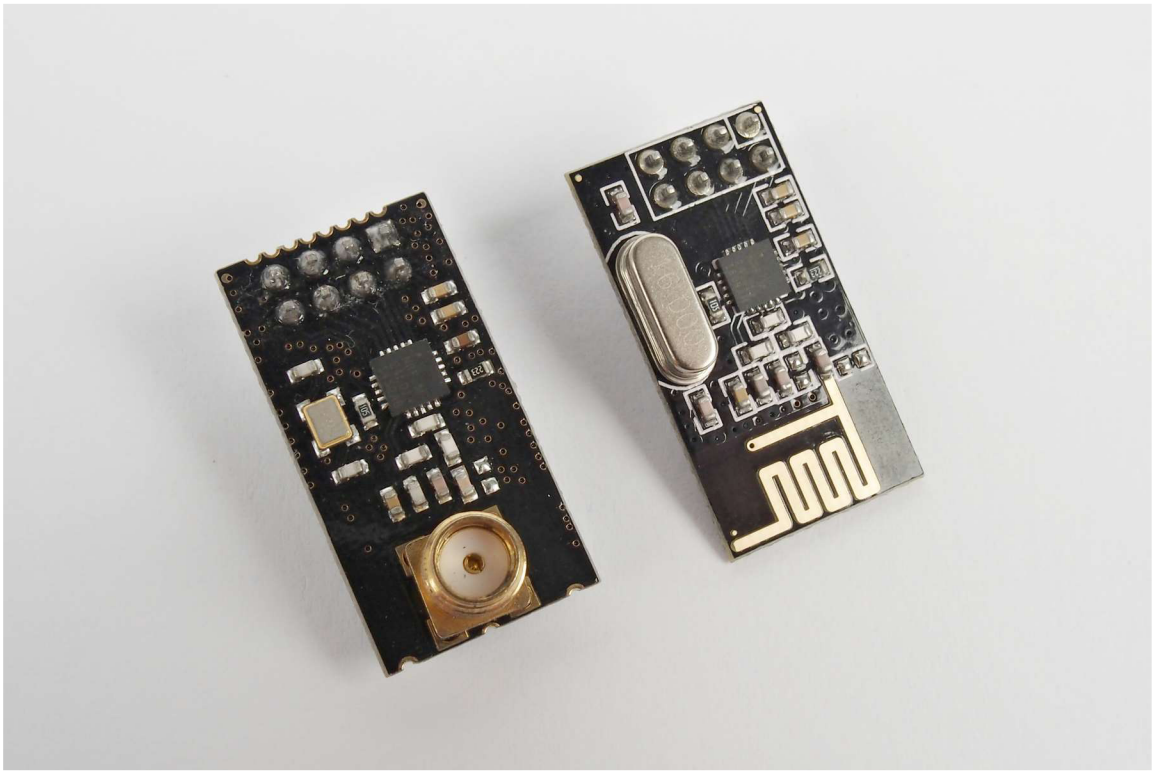


Obrázek 3.7: Deska Wemos D1 s čipem ESP8266. Obrázek přejat z [26].

Jednou z možností je například modul nRF24L01/nRF24L01+ od firmy Nordic Semiconductor. Původně byly tyto čipy určené do bezdrátových periferií (klávesnice, myši) a v současné době již nejsou společností Nordic Semiconductor podporované. Ale díky jejich kvalitám je pro Arduino dostupná komunitou vytvořená knihovna a dají se tedy bez problémů používat.

Samotný modul pracuje v pásmu 2.4 GHz, tedy ve stejném pásmu jako WiFi, což kvůli zahlcení tohoto pásma poněkud snižuje dosah. Ten se pohybuje od jednotek metrů při zabudované anténě a průchodu signálu překážkami jako jsou zdi, až po nízké stovky metrů při maximálním výkonu, externí anténě i zdroji napájení a přímé viditelnosti. Teoretické maximum je poté zhruba tisíc metrů.

Další moduly pracují například v pásmu 433 MHz nebo 868 MHz. Nižší frekvence znamená lepší dosah a prostupnost signálu překážkami, na druhou stranu ale také horší rychlosti, což ovšem není u mikrokontrolerů a podobných zařízení typicky hlavním problémem.



Obrázek 3.8: Modul nRF24L01. Obrázek přejat z [47].

Kapitola 4

Návrh tutoriálu a robota Trilobot

V této kapitole bude probrán návrh tutoriálu tak, aby pokrýval všechna nezbytná témata a zároveň splňoval požadavky na modularitu. Modularita přitom bude dvojího druhu: tutoriál půjde přizpůsobit jak úrovni studenta/studentů, tak času, který bude pro tutoriál k dispozici. Zároveň zde bude probrán návrh robota Trilobot jak po hardwarové, tak softwarové stránce.

Obecnými cíli tutoriálu bylo zaplnit „prázdné místo“, které bylo identifikováno v závěru sekce 2.3. Tedy vytvořit tutoriál, který by uživatele provedl oživením robota a vysvětlil jim alespoň některé oblasti programování Arduina a robotiky, ideálně v takové míře, aby byl student po absolvování tutoriálu schopen na získaných dovednostech dále stavět.

Některými prostředky k dosažení těchto cílů jsou například kompletní přehledy příkazů, které se v tutoriálech běžně nevyskytují, nebo odkazy na dokumentace jednotlivých senzorů i s návody, jak a kde v nich najít požadované informace[1].

4.1 Návrh tutoriálu

Před tvorbou slajdů proběhl jednoduchý konceptuální návrh tutoriálu, který bude popsaný dále.

Hlavním smyslem slajdů je poskytnout osnovu lektorovi v situacích, kdy je tutoriál používán skupinou více lidí, kteří se na něm mají učit společně. Další situací, kdy je možné využít prezentace, je pro pro rychlé sdělení informací, například na různých akcích pro veřejnost. Z toho vyplývá, že slajdy by měly poskytovat právě onu pevnou osnovu a všechny nezbytné informace v nezbytné míře. Na druhou stranu, případy použití předpokládají, že slajdy bude promítat lektor, který platformě Arduino rozumí a který případně může poskytnout pomoc, takže informace nemusí být příliš detailní nebo vyčerpávající.

V konceptu jsem tedy tutoriál rozčlenil na tři hlavní části nazvané„Úvod“, „Před začátkem tutoriálu“ a „Tutoriál“, které budou podrobněji popsané dále. Část „Tutoriál“ poté bude dále členěna do tzv. „modulů“, tedy sekcí, které se budou věnovat konkrétnímu tématu, například měření intenzity světla daným senzorem nebo vytvoření detektoru otřesů.

Samotný modul je přitom základní jednotkou tutoriálu a jeho cílem je vždy probrat a vysvětlit jednu izolovanou funkcionalitu – například zprovoznění jednoho daného senzoru nebo například vytvořit funkci pro jízdu rovně. Moduly jsou na sobě prakticky nezávislé a je

[1] Odkazy na dokumentaci jednotlivých modulů a senzorů chybí v drtivé většině tutoriálů a studenti, neznaje, že takový zdroj informací vůbec existuje, jsou odkázání na dlouhé hodiny hledání po diskuzních fórech. Proto i kdyby si studenti odnesli pouze povědomí o tom, že existuje něco jako dokumentace jednotlivých komponent, mělo by toto rozhodnutí smysl.

tedy možné je probírat v jakémkoli počtu a pořadí. Tím je docílena určitá časová flexibilita tutoriálu. V potaz je brána i časová modularita a jeden modul by tedy měl trvat řádově desítky minut.

Každý modul bude mít svou strukturu, která se napříč moduly nebude příliš měnit:

- Na začátku každého modulu je vysvětlen modul Trilobota, případně funkcionalita, která bude implementována. Často jsou zde také vysvětleny úkoly v přirozeném jazyce.
- Následuje sekce „Úkoly“, která už obsahuje jednotlivé kroky, vedoucí k úspěšné implementaci.
- Jako další modul obsahuje kostru, shodnou s kostrou funkce v operačním kódu Trilobota. Tato kostra může sloužit pro vysvětlení některých jevů a podobně.
- Nyní následuje samotný návod krok po kroku, jak se dostat k funkční implementaci. Jelikož se jedná o tutoriál, ve slajdech je takto popsané kompletní řešení.
- Jako poslední je ve slajdech uvedena i kompletní, doplněná funkce.

Dále jsem se rozhodl, že nad rámec zadání vytvořím pro českou mutaci slajdů i doprovodný text, který bude detailněji popisovat probíranou problematiku. Jelikož doprovodný text není nezbytnou součástí práce, podrobnější informace jsem zařadil na konec této sekce.

4.1.1 Modularita

Výše již byla popsaná modularita z hlediska času. Ve stručnosti, tutoriál je rozdělený na moduly, které jsou samy o sobě malými tutoriály, zaměřenými na zprovoznění konkrétního senzoru nebo jiné periferie robota.

Modularita z pohledu předpokládané úrovně znalostí studentů bude řešena jiným způsobem. Prvotní myšlenka počítala s tím, že by slajdy, stejně jako doprovodný text (4.1.3), obsahovaly různé pokročilé sekce pro využití pro pokročilejší studenty. Toto řešení bylo ale záhy opuštěno, jelikož by slajdy působily zmatečně a nebylo byjasné, která informace patří k jakému tutoriálu.

Návrh tedy počítá s vytvořením tří oddělených sad slajdů na úrovních, které odpovídají znalostem a schopnostem studentů druhých stupňů základních škol, studentů středních škol a studentů univerzit, případně zájemcům z řad technických nadšenců. Výchozí verzí bude ta pro studenty vysokých škol a zbylé dvě vzniknou zjednodušením této výchozí verze. Přesné informace o zjednodušení budou uvedeny v následující kapitole, návrh ale počítá s tím, že verze pro žáky základních škol bude oproti dvěma následujícím verzím zkrácena. Důvodem je to, že pro žáky středních škol to dost možná bude první setkání s programováním robotů a delší tutoriál by byl tedy zbytečně dlouhý.

4.1.2 Podrobnosti návrhu

Níže uvádím podrobněji popsané jednotlivé části tutoriálu, jejich význam a přibližný obsah.

Úvod má za cíl seznámit uživatele s robotem Trilobot a samotným tutoriálem, tedy vybavení Trilobota atp. Případně umožňuje říci lektorovi pár slov o sobě a podobně.

Před začátkem tutoriálu je jakýmsi prvním modulem, u kterého se předpokládá, že bude probrán vždy, ovšem ne nezbytně v plném rozsahu, záleží na úrovni posluchačů. Skládá se z šesti částí, pojmenované „Poznámky“, „Reprpduktor“, „Dispej“, „I^2C “, „Arduino IDE a připojení Trilobota“ a „Úvod do programování Arduina a OOP“. Znalosti ze všech těchto

částí jsou nezbytné ke zvládnutí tutoriálu a proto jsou vyčleněny mimo samotné moduly, aby bylo jasné, že se nejedná o „standardní“ modul (o modulech bude řeč dále).

V části „Dispej“ je popsán I^2C displej a knihovna `display.h`. Knihovna `display.h` je wrapperem nad knihovnou `LiquidCrystal_I2C.h` a jejím účelem je maximální zjednodušení výpisu hodnot na displej – pro práci s displejem přes tento wrapper postačují dvě funkce.

Část „I^2C “ se věnuje I^2C sběrnici. řehledně a na jednom místě jsou popsány všechny základní příkazy pro práci s touto sběrnicí a studenti se k nim mohou velmi jednoduše vracet.

Poslední dvě části („Arduino IDE a připojení Trilobota“ a „Úvod do programování Arduina a OOP“) se věnují samotným základům práce s Arduinem a Trilobotem (připojení Trilobota k počítači a otestování spojení) a také základnímu programování Arduina a základům OOP (operační kód Trilobota je objektově orientovaný, vizte další sekci).

Tutoriál již obsahuje samotné moduly, o kterých bylo diskutováno výše. V tuto chvíli se předpokládá následující rozdělení:

- Modul o měření vzdálenosti, případně tři moduly pro každý ze senzorů (HC-SR04, SRF-08 a Sharp) se společným úvodem.
- Modul o ovládání motorů, pravděpodobně rozdělený do dvou modulů, pro jízdu rovně a pro zatáčení.
- Moduly, věnující se minIMU-v3. V tuto chvíli se předpokládá hlavně praktická aplikace vždy jen jednoho z trojice senzorů (gyroskop, akcelerometr, magnetometr)

4.1.3 Doprovodný text

Jak již bylo zmíněno výše, česká mutace bude obsahovat kromě slajdů i doprovodný text. Tento text svou osnovou do jisté míry kopíruje slajdy, v něčem je i pozměňuje a doplňuje (vizte podsekci 4.1.3).

Hlavním cílem doprovodného textu je rozšířit informace o jednotlivých modulech a umožnit použití tutoriálu i v prostředí bez lektora. Další výhodou je i možnost sdělit některé zajímavé, ale nikoli stěžejní informace. Také zde bylo možné probrat i oblasti, které se přímo nehodí do slajdů, ale doplňují téma robotiky jako celku.

Nepředpokládá se, že by byl doprovodný text přímo používán během tutoriálu, ale mohl by poskytovat zajímavé informace pro studenty, které by tutoriál zaujal a chtěli se dozvědět o Trilobotovi, případně i robotice nebo Arduinu, podrobnosti, které nemohly být z úsporných důvodů umístěny do slajdů.

Rozdíly doprovodného textu oproti slajdům

Kvůli logickému dělení textu je tato podsekce v návrhové části práce, vznikala ovšem teprve po první části implementace, kdy již bylo jasné, které informace budou v doprovodném textu rozváděny a které nikoli.

Oproti slajdům je v doprovodném textu několik kapitol navíc: kapitola o robotech a robotice, která mimo jiné ukazuje i některé roboty, vyvíjené na FIT VUT.

Další důležitou kapitolou je kapitola zahrnující komunikaci mezi více deskami. V této kapitole jsou vysvětleny některé ze základních principů komunikace, které se shodují s těmi v teoretické části této práce (3.4.4).

Do slajdů toto téma zahrnuto nebylo, protože Trilobot bohužel nedisponuje žádným modulem, schopným komunikovat a jednalo by se tedy pouze o teorii bez možnosti praktického vyzkoušení. V doprovodném textu je ale toto téma zpracováno relativně podrobně a dává k dispozici i základní přehled příkazů pro vybrané často používané moduly. V budoucnu může také tato kapitola sloužit jako základ při rozšiřování Trilobotovy funkcionality.

4.2 Trilobot

Níže bude stručně shrnutý návrh přestavby Trilobota a vytvoření ovládacího kódu, na kterém se bude prakticky realizovat i tutoriál. Jelikož se návrh skládal ze dvou velmi odlišných části, budou probrány odděleně, i když jejich návrh (a později i vývoj) probíhal paralelně.

4.2.1 Návrh hardware

V této sekci proberu návrh hardware Trilobota - tedy propojení senzorů, způsob jejich uchycení a v neposlední řadě i samotný přehled použitých senzorů.

Nejdříve bylo nutno rozhodnout, jakým způsobem a na jakých místech budou moduly upevněny a jak budou komunikovat s deskou Arduino. Výsledkem bylo schematické zapojení, které můžete vidět na obrázku 4.1.

V původním zadání měl být Trilobot osazen pouze moduly Sharp, SRF-08, minIMU-v3, motory Faulhaber a deskou Sabertooth 2x5. Hlavně během práce na semestrálním projektu byl návrh pozměněn a rozšířen o další moduly: senzor HC-SR04 a I^2C LCD displej 16x2.

Senzory HC-SR04 a SRF-08 spolu demonstrují dva možné přístupy k senzorům: zatímco senzor SRF-08 je zástupcem dražší třídy ultrazvukových senzorů s podrobným nastavením (které v tutoriálu ani není využito), senzor HC-SR04 je naproti tomu levným a oproti SRF-08 i „primitivním“ senzorem, který měří vzdálenost pouze nepřímo a výpočet je poté realizován na straně Arduina z času mezi vysláním a příjmem zvukové vlny.

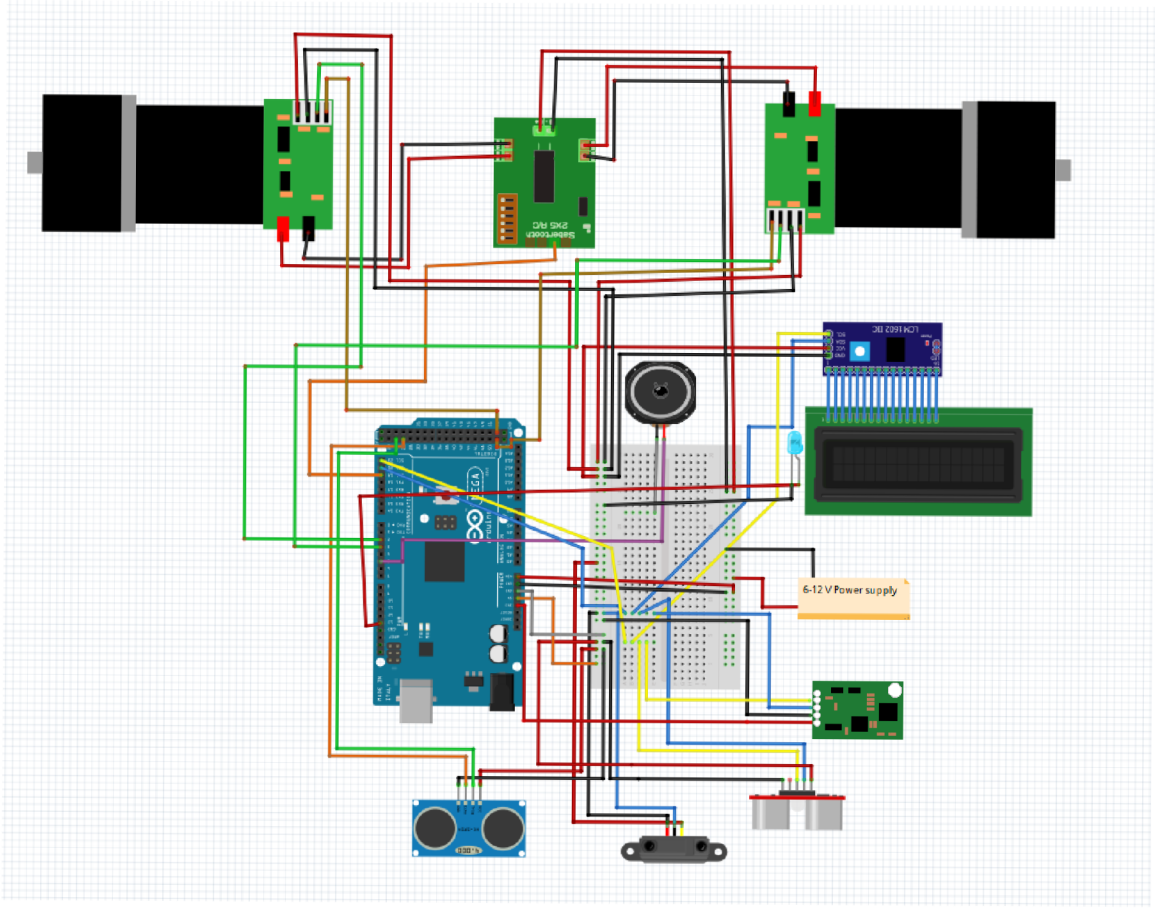
Studenti mohou sami porovnat výhody a nevýhody obou senzorů a pokud budou sami v budoucnu tvořit nějaký projekt, mohou se sami rozhodnout, který typ senzoru by lépe vyhovoval jejích cílům.

Displej byl přidán jako zobrazovací zařízení, jelikož původně se počítalo pouze se zasíláním údajů přes seriovou linku do počítače. Tento přístup se mi ale zdál jako nevhodný, jelikož Trilobot je ze své podstaty mobilní; USB kabel k počítači by tedy musel být dlouhý několik metrů nebo by musel student za Trilobotem s počítačem chodit, pokud by se jednalo například o notebook. Přidání displeje tento problém do jisté míry vyřešilo.

Níže budou velmi stručně popsány jednotlivé senzory, s jejichž montáží se počítá. Kromě popisu senzoru bude popsán i návrh umístění a uchycení senzoru. Obecně se počítá s tím, že pro uchycení budou mnou vytvořeny úchyty na míru, které budou následně vytištěny na 3D tiskárně.

SRF-08

SRF-08 je ultrazvukový měřič vzdálenosti kombinovaný se světelným senzorem. Komunikuje prostřednictvím I^2C sběrnice. Podrobnější informace o jeho použití lze nalézt v dokumentaci výrobce [3]. Senzor bude upevněn v přední části Trilobota, při pohledu zpředu vlevo v držáku, který umožňuje otáčení doleva a doprava.



Obrázek 4.1: Návrh zapojení robota Trilobot, odpovídající skutečnosti. Obrázek byl vytvořen v programu Fritzing: https://fritzing.org/home/.

Obrázek 4.2: Senzor SRF-08

HC-SR04

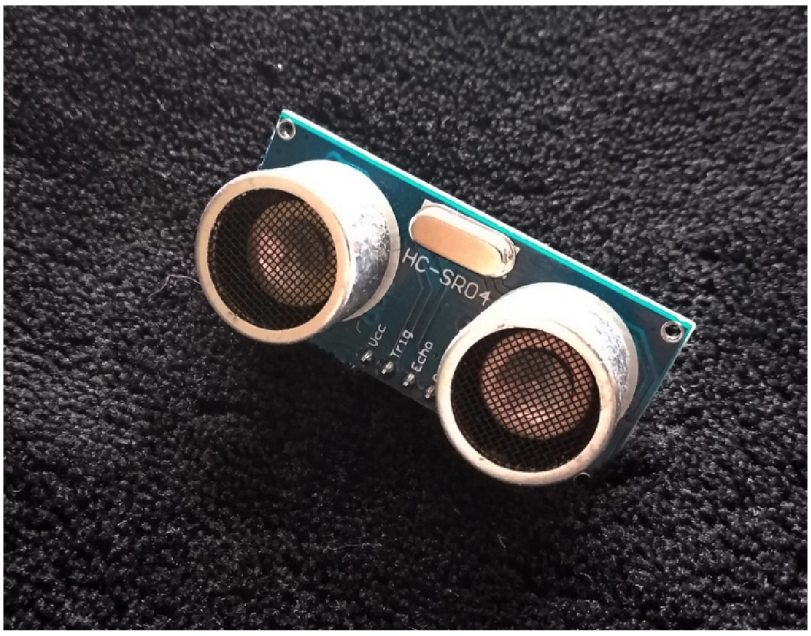
HC-SR04 je stejně jako předchozí senzor ultrazvukový měřič vzdálenosti. Na rozdíl od SRF-08 ale nekomunikuje po žádné sběrnici ale aktivuje se přímo aktivačním vodičem. Naměřená hodnota v milisekundách je odečítána na straně Arduina jako délka impulzu na jiném vodiči. Jeho hlavní předností je mnohem nižší cena. Na druhou stranu ale není tak přesný. Výrobce Sparkfun k němu dodává i dokumentaci [6]. Senzor bude umístěný v přední části Trilobota, při pohledu zepředu vpravo v držáku, umožňujícím pohyb doleva a doprava.

Sharp (GP2Y0A41SK0F)

Poslední ze skupiny senzorů měřících vzdálenost. Od předchozích dvou se liší hned několika věcmi. Především pracuje na principu odrazu infračerveného světla, ne zvuku. Vzdálenost se poté nepočítá jako délka od vyslání k přijetí vlny (tento čas by byl ve většině případů extrémně krátký a pravděpodobně neměřitelný pro Arduino), ale k jeho výpočtu se používá hodnota napětí na datovém vodiči. Senzor je umístěn vpředu Trilobota uprostřed. Pokud jsou všechny tři senzory vzdálenosti správně nastaveny, poskytují relativně přesné informace o prostoru před Trilobotem.

I^2C displej 16x2

Jedná se o standardní displej 16x2, tedy dvouřádkový se 16 znaky na každém řádku. Pro ušetření vodičů je k němu připájené rozhraní pro komunikaci přes I^2C sběrnici. Slouží pro zobrazování naměřených údajů a případně pro podrobnější zpětnou vazbu. Displej je umístěn ve vytištěném pouzdru na přední straně Trilobota. V pouzdru se zároveň nachází LED, připojená na vestavěnou LED Arduina.



Obrázek 4.3: Senzor HC-SR04

Motory Faulhaber 16002, řídící deska Sabertooth 2x5

Pohonná soustava robota Trilobot se skládá ze dvou motorů Faulhaber 16002 a řídící desky Sabertooth 2x5. Deska Sabertooth je schopná pracovat v mnoha různých režimech, pro robota Trilobot je využito zjednodušené ovládání po sériové lince („Simplified Serial“). Motory Faulhaber poskytují i zpětnou vazbu v podobě enkodérů, podle kterých se dají určit jejich otáčky.

Motory jsou umístěné po stranách nad hnanými koly, Řídící deska Sabertooth je umístěna v zadní části Trilobota mezi motory, Fixována oboustrannou lepicí páskou. Toto řešení bylo zvoleno z toho důvodu, že nebylo nutné vyvrtávat žádné otvory ani jinak invazivně zasahovat do kostry Trilobota.

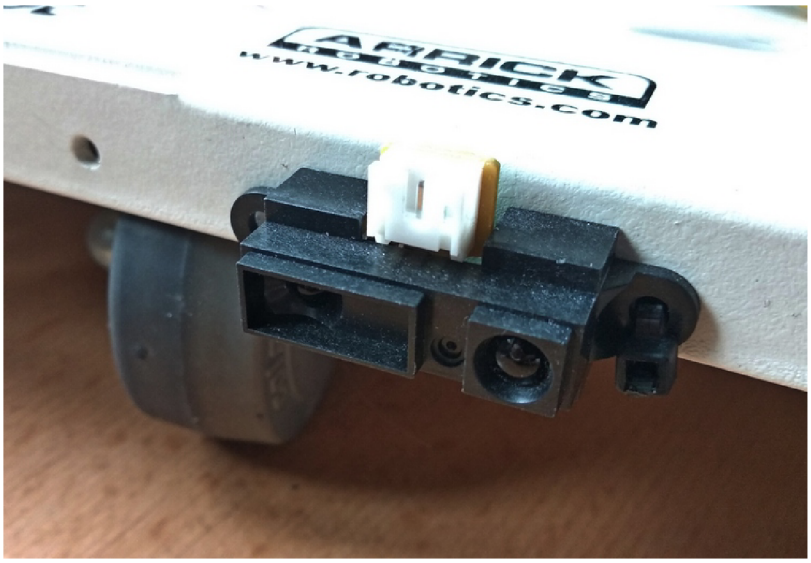
minIMU-v3

Tento senzor v sobě skrývá akcelerometr, gyroskop a magnetometr. S Arduinem komunikuje přes sběrnici I^2C . Jeho uchycení je vyřešeno „věžičkou“ s montáží v horní části Trilobota. Toto řešení je zvolení z toho důvodu, že magnetometr, jedna ze součástí senzoru minIMU-v3, je rušen tělem Trilobota a pokud by byl přimontován blíže, rušení by bylo příliš velké a nešlo by kompenzovat.

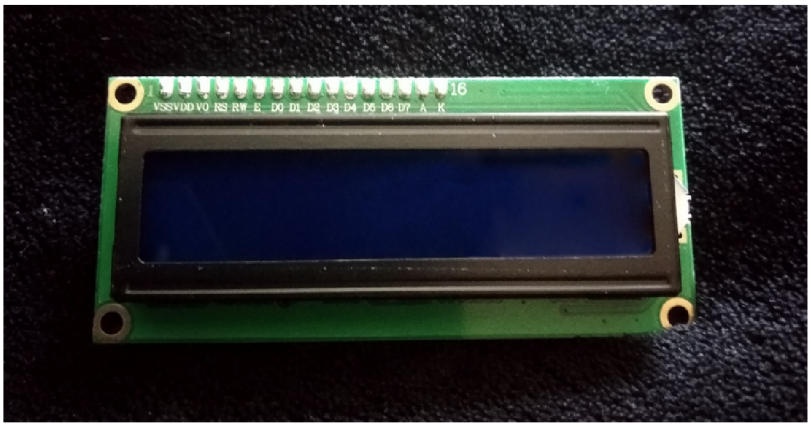
Reproduktor

Jedná se o obyčejný piezoelektrický reproduktor o neznámém výkonu. Primárním účelem je akustická zpětná vazba některých akcí, je ale také plně programovatelný a lze na něm zahrát libovolnou melodii. Je umístěn na levé straně při pohledu zprava.[2]

[2]Během implementace nakonec nebyl reproduktor využit, jelikož by v místnosti s více lidmi dělal zbytečný hluk.



Obrázek 4.4: Senzor Sharp 1994 (GP2Y0A41SK0F)



Obrázek 4.5: I^2C displej 16x2

4.2.2 Montáž a zapojení

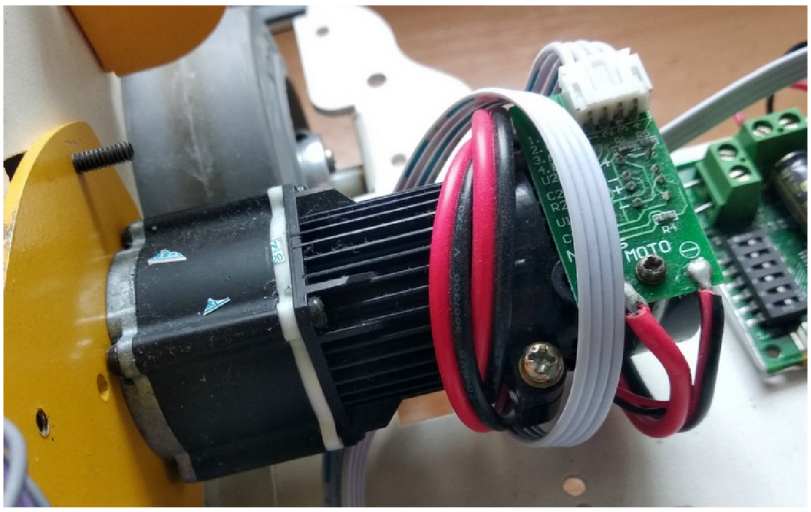
Zde znovu shrnu na jednom místě, jakým způsobem byly senzory osazeny a jak konkrétně byly připojeny k Arduinu.

Pro montáž senzorů byly vytvořeny a vytištěny držáky senzorů SRF-08, HC-SR04, Sharp GP2Y0A41SK0F a montáž senzoru minIMU-v3. Na 3D tiskárně byl také navrhnut a vytištěn kryt displeje.

Během práce s magnetometrem (součást senzoru minIMU-v3) se objevil problém. Magnetometr je rušený velkým množstvím kovu, ze kterého je Trilobot vyroben, a proto byl dodatečně namontován na malé „věžičce“, jak je vidět na fotografii **??**. Tato skutečnost tedy byla do návrhu přidána zpětně.

Senzory, které to umožňují[3], jsou připojeny přes sběrnici I^2C . Toto řešení umožnilo zredukovat počet kabelů (zvlášť u displeje).

[3]SRF-08, minIMU-v3, displej



Obrázek 4.6: Motor Faulhaber

Deska Sabertooth 2x5 pro ovládání motorů komunikuje s Arduinem přes seriovou linku `Serial2`, a to pouze v jednom směru – od Arduina k desce (jedná se o mód `Simplified Serial`, další módy dostupně v manuálu [4].

Enkodéry motorů Faulhaber mají sice zapojené oba kanály, ale hned z několika důvodů se v kódu používá pouze jeden z každého enkodéru. Hlavním důvodem je omezený počet pinů, které na desce Arduino Mega podporují přerušení. Vedlejšími důvody jsou například zjednodušení (informaci o směru Kód v současné chvíli nepotřebuje a v tutoriálu by použití obou kanálů nepřineslo žádné výhody).

Napájení

Trilobot bude dvě možnosti napájení: interní přes Arduino a externí přes přídavný konektor.

Interní napájení je realizováno přes 5V a 3.3V výstup Arduina, do kterého energii přivádí USB kabel z hostitelského počítače. Arduino je tak schopné napájet všechny senzory, ovšem deska Sabertooth a na ni napojené motory zůstávají vypnuté.

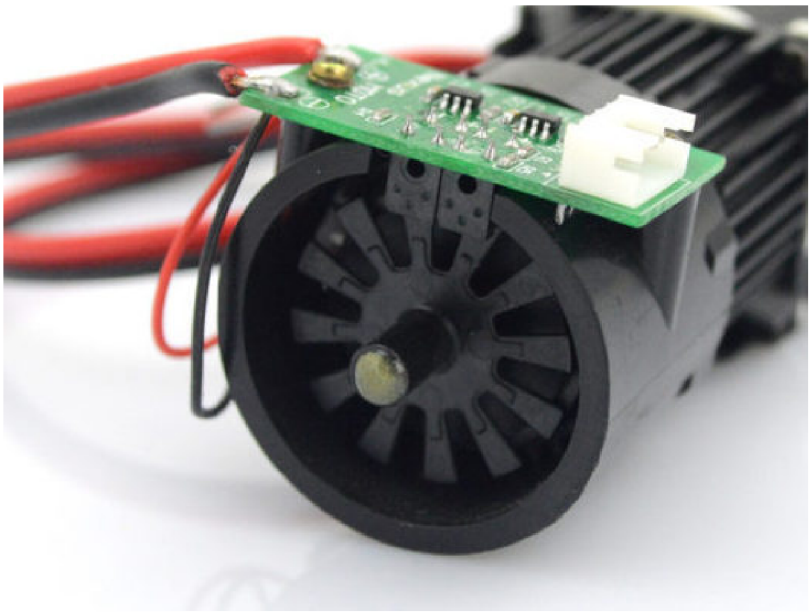
Pro aktivaci motorů a desky Sabertooth je nutné Trilobota napájet pomocí externího zdroje energie o napětí 7-12 V. Tento zdroj může být připojen pomocí 5.5/2.1mm kulatého konektoru. Tuto energii poté využívá jak deska Sabertooth, tak i Arduino, které je ke zdroji napojeno přes piny Vin a GND.

4.2.3 Návrh software

Původní návrh počítal s neobjektovým návrhem a pouze jednotlivými „sketchi“ pro jednotlivé moduly tutoriálu. Krátce po sestavení Trilobota se ale ukázalo, že by mj. byl potenciál robota Trilobot zbytečně nevyužitý a návrh byl kompletně změněn a přetvořen do objektového návrhu. Tento změněný návrh je uveden i zde.

Důvody pro změnu návrhu

Jak již bylo řečeno, v pozdější fázi se ukázaly nesporné výhody objektového návrhu, které nakonec zapříčinily refaktorizaci již napsaného kódu do objektové orientace.



Obrázek 4.7: Enkodér motoru Faulhaber

Hlavní důvod byl edukativní. Trilobot se skládá z modulů – objektů reálného světa – se svými vlastnostmi a schopnostmi. Proto se zdá logické, aby kód, který se stará o ovládání daného objektu reálného světa, byl jeho co nejvěrnějším modelem v kódu – tedy objektem tak, jak ho chápe OOP.

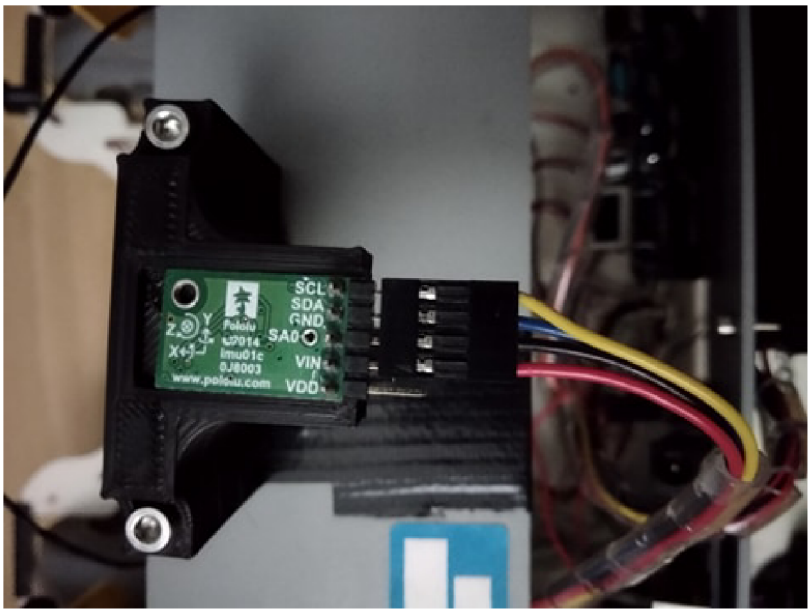
Předpoklad je takový, že si studenti lépe spojí objekt reálného světa – například senzor vzdálenosti – a objekt v kódu se stejným jménem a lépe tak pochopí, co daný kód dělá. Pochopení základů OOP je už jen bonusem.

I když se tento názor zdá logický a některé studie potvrzují pozitivní účinek v opačném směru – tedy že využití robotů pomáhá ve výuce OOP [37] – dosud neproběhla studie dokazující opak. V budoucím testování bude tedy třeba klást důraz na zjištění, zda se studenti chovají podle předpokladu, tedy že jsou schopni si spojit objekty reálného světa a objekty napsané v programovacím jazyce.

Dalším důvodem bylo zjednodušení funkcí a omezení duplikujících se názvů funkcí, lišících se jen názvem senzoru – tento problém vznikal například u senzorů vzdálenosti (všechny senzory umožňovaly měřit vzdálenost a název funkce byl tedy prakticky totožný) a často byl velmi matoucí. Výsledný kód je nyní přehlednější a je lépe vidět, která data a které funkce patří jakému konkrétnímu senzoru.

Upravený návrh tedy kód rozdělil do jednotlivých tříd, kde každý objekt dané třídy ovládal sobě příslušný senzor nebo modul. Objekty jsou v kódu již inicializované a připravené k použití. I když se nejedná o návrhový vzor Jedináček (V budoucnu může být možné rozšířit Trilobota i o několik senzorů stejného typu), v současné chvíli se nepředpokládá, že by bylo třeba vytvářet jiné než předpřipravené objekty. Níže v přehledu uvádím podrobný popis jednotlivých objektů.

Jak bude následně vidět z implementace jednotlivých tříd dále, kostry metod, které se mají doplňovat v tutoriálu, mají i jakési „referenční předobrazy“. Tyto funkce jsou plně funkční a často obsahují např. podrobnější kontrolu parametrů, která v tutoriálu byla zbytečná.



Obrázek 4.8: minIMU-v3

Účelem těchto funkcí je zajistit, aby byl tutoriál plně použitelný i mimo tutoriál. Pokud se tedy z kódu odstraní kostry funkcí, stále zůstane plně funkční kód, který může být dále rozšiřován.

Návrh ovládacího kódu

Podrobný návrh (například diagram tříd) jsem před samotnou implementací nevytvářel, jelikož samotný kód vznikal na základě potřeb tutoriálu. Níže proto shrnuji alespoň některé body, kterých jsem se během psaní kódu držel.

Celý kód je komentovaný ve standardu doxygen v českém jazyce a lze tedy vygenerovat i programovou dokumentaci. Stejně tak jsou komentovány i některé části kódu, které nemusí být na první pohled jasné.

Kód je psaný anglicky, kromě výjimky níže (české názvy funkcí pro doplnění do tutoriálu). Důvod, proč není celý tutoriál psaný česky, je ten, že programování probíhá výhradně v angličtině a je dobré na to studenty tímto upozornit a nechat je si (pasivně) zvykat na anglický kód, pro ulehčení opatřený českými komentáři.

Samotné metody určené pro tutoriál jsou odlišeny od zbytku kódu jak v komentářích, tak názvem, který je psán v českém jazyce. To na první pohled pomáhá s identifikací takové funkce, která má být doplňována. Nesourodost kódu je tedy záměrná.

Ovládací kód jako celek je psán tak, aby i bez doplněných tutoriálových funkcí byl plně funkční. Ovládací kód je tedy možné použít i bez spojitosti s tutoriálem, například pro samostatné aktivity studentů.

Kapitola 5

Implementace

Implementace kódu pro Trilobota proběhla podle návrhu uvedeném v kapitole 4 v jayce Wiring/C++. Kód se dá přeložit a nahrát do robota na libovolném operačním systému, který podporuje Arduino IDE ve verzi 1.8.10 a vyšší.

V prvním kroku byl sestaven prototyp nové verze Trilobota[1], na kterém se dal prakticky vyvíjet tutoriál. Poté byl implementován kód a vytvořen tutoriál pro modul, věnující se měření vzdálenosti. Po implementaci tohoto modulu proběhlo pilotní testování (vizte 6), jehož cílem bylo zjistit, zda implementovaný návrh je v koncepci správný. Poté, co výsledky potvrdily, že návrh tutoriálu splňuje základní cíle (srozumitelnost aj.), byly implementovány i ostatní části.

5.1 Detailní popis jednotlivých tříd

Níže uvádím podrobný popis jednotlivých implementovaných tříd i s jejich vlastnostmi.

Struktura hlavního souboru `trilobot.ino`

V hlavním souboru jsou nejprve deklarovány globální proměnné všech potřebných objektů (vizte dále). Všechny objekty jsou později inicializovány ve funkci `setup()`. V této funkci je dále zahájena komunikace po seriové lince pro případnou komunikaci s hostitelským počítačem a komunikace přes sběrnici I^2C . Funkce `loop()` je standardně prázdná, výjimku tvoří pouze podmíněný blok kódu pro pravidelné vykonávání příkazů (vizte dále).

V souboru `trilobot.ino` lze pomocí makra `WITH_20HZ_LOOP` zapnout i podmíněný blok příkazů ve funkci `loop()`, který se bude pravidelně vykonávat jednou za $\frac{1}{20} + t$ s, kde t je čas vykonávání hlavní smyčky.

Princip smyčky tvoří periodické přerušení s periodou 50 ms, realizované externí knihovnou `TimerThree` [39]. Během tohoto přerušení je nastaven flag `isTime` na hodnotu `true` a podle hodnoty tohoto flagu je případně vykonán podmíněný blok příkazů, kde se ve výchozím stavu aktualizuje poloha trilobota.

Třída `SRF08`

Tato třída ovládá senzor srf08 a umožňuje provádět měření vzdálenosti a intenzity osvětlení. Třída obsahuje tyto metody:

[1] Senzory byly přechodně přilepeny lepicí páskou atp.

- `void set_measurement(byte unit)`: funkce slouží pro zahájení měření, kdy odešle do modulu srf-08 informaci o požadované jednotce (k dispozici jsou centiemtry, palce nebo milisekundy, které se dají nastavit příslušnými makry `CM, INCH, MS`). Funkce dále čeká 100 ms, aby senzor mohl provést měření.
- `int get_distance(byte unit)`: Funkce vyšle na senzor pomocí předchozí funkce pokyn k měření, přečte hodnotu z příslušných registrů senzoru a vráti ji.
- `byte get_light_intensity()`: funkce zahájí měření v libovolné jednotce[2], po jeho ukončení přečte hodnotu světelné intenzity a tu vrátí.
- `byte zmer_svetlo()` a `int zmer_vzdalenost()` jsou kostry funkcí, které budou doplněny v tutoriálu.

Třída `HCSR04`

Třída implementuje ovládací logiku pro senzor HC-SR04. Obsahuje tyto metody:

- `long get_microseconds()`: funkce vrací pouze časv ms, který uplynul od odeslání po příjem zvukové vlny.
- `long get_distance()`: Funkce vrací vzdálenost v cm od překážky. K převodu je použita fixní rychlost zvuku ve vzduchu 340 m/s, při různé teplotě nebo v jiném prostředí by tedy mohla mít funkce odlišné výsledky.
- `long zmer_vzdalenost()`: kostra funkce pro doplnění v tutoriálu.

Třída `Sharp`

Tato třída ovládá senzor Sharp GP2Y0A41SK0F. Obsahuje pouze dvě veřejné metody:

- `get_distance()`: Funkce vrací vzdálenost v centimetrech. Přepočet z napětí na datovém pinu probíhá pomocí lineárního splajnu, kdy byla závislost naměřeného napětí na vzdálenosti z dokumentace [5] rozdělena na několik úseků, které se daly aproximovat přímkou. Experimentálně bylo ověřeno, že současná aproximace vrací validní výsledky[3].
- `zmer_vzdalenost()`: Kostra funkce pro doplnění v tutoriálu.

Třída `Display`

Třída je wrapperem kolem knihovny `LiquidCrystalq_I2C` [35]. Slouží pro co nejjednodušší použití displeje, ideálně takové, aby ho byl člověk, který s Arduinem nikdy nepracoval, schopný používat do jedné minuty. Ovládání displeje bylo tedy maximálně zjednodušeno a výsledkem jsou pouze tři funkce pro ovládání:

- `void print_first_line()` a `void print_second_line()`: Tyto funkce smažou daný řádek a vytisknou jednu hodnotu libovolného tisknutelného datového typu[4].

[2] Senzor srf-08 neumí změřit pouze intenzitu světla, ale vždy je třeba vyvolat i měření vzdálenosti. Tato hodnota je poté nevyužita.

[3] Odchylka se pohybovala zhruba kolem 10 %, pro Trilobota je toto dostatečná hodnota, pokud by ovšem byl senzor používaný v kritičtějších aplikacích, bylo by třeba zpřesnit lineární splajn, použít jiný typ aproximace či najít takovou funkci, která co nejlépe odpovídá závislosti z dokumentace.

[4] Podporované jsou mj. byte, int, float, double, char, string, tedy všechny základní datové typy.

- `void clear()` Funkce smaže celý displej.

Použití displeje je tedy velmi omezeno, ale svůj účel plní velmi dobře.

Třída `Speaker`

Třída slouží pro ovládání vestavěného reproduktoru. Jelikož reproduktor dělá poměrně velký hluk, nebyl pro něj vytvořen tutoriálový modul. Ovšem pro ukázky, například na Dnech otevřených dveří, by se reproduktor mohl hodit a proto je jeho programová podpora implementována. Kód byl částečně přejat, případně byl inspirován z [18].

- `void enable()` a `void disable()`: tyto funkce zapínají, resp. vypínají reproduktor. Pokud je reproduktor zapnutý, ale nehraje žádný tón, vydává nepříjemné pískání.
- `void imperial_march()`: Funkce zahraje část skladby „Imperial MArch“ z Hvězdných válek jako demonstraci reproduktoru.
- `void beep(int note, int duration)`: Funkce zahraje vybranou notu o dané délce. Přehled not jek dispozici v `spekaer.h`.

Třída `Motors`

Třída slouží k ovládání motorů Faulhaber přes desku Sabertooth. Pro zpětnou vazbu o ujeté vzdálenosti využívá enkodéry, konkrétně jeden kanál pro každý motor.

Tento jeden kanál z každého ze dvou enkodérů je napojen na piny podporující přerušení a ve chvíli, kdy dojde ke změně hodnoty na vodiči (přechod z log. 0 do log. 1) je vyvoláno přerušení, pokud je povoleno. K povolení přerušení dochází pouze ve chvíli, kdy se motory hýbou, tedy od doby těsně před odesláním do doby těsně po zastavení motorů.

Obsahuje metody nacházející se níže:

- `void move(int distance, int speed)`: Funkce zajišťuje jízdu rovně. Pokud je funkci zadán pouze jeden parametr, ten je považován za rychlost a pohyb musí být ukončen explicitním zavoláním metody `stop()`. Pokud jsou zadány oba parametry, motory se pohnou o danou vzdálenost v cm.
- `void turn(int angle, int speed)`: Funkce slouží pro zatáčení o dané úhel. Implementováno je zatáčení, kdy jedno kolo stojí na místě a druhé kolem něj opisuje kružnici.
- `void circle(int diameter, int speed)`: Funkce by měla umožnit jezdit nepřetržitě v kruhu o daném poloměru. Tato funkce bohužel nebyla implementována.
- `stop()`: Funkce zastaví oba motory.

Třída `Gyroscope`

Třída má za cíl ovládat gyroskop z modulu minIMU-v3. Součástí třídy je i metoda pro implementaci jednoduchého senzoru otřesů/vibrací.

Gyroskop má také k dispozici registr s hodnotou teploty. Tento teplotní registr je ovšem velmi stroze dokumentovaný a v dokumentaci například není uveden převod ze surové hodnoty na teplotu ve stupních Celsia. Tento převod byl získán reverzním inženýrstvím a experimenty.

Všechny tři třídy obsluhující části modulu minIMU-v3 využívají datový typ `vector`, o kterém bude řeč později v kapitole Implementace.

- `get_raw_data()`: Získává surová data z registrů senzoru a vrací je.
- `get_angluar_velocity()`: Nejprve získá surová data a ta poté převede na hodnoty úhlového rychlosti ve třech osách X,Y,Z. Převod je závislý na konkrétních nastavení citlivosti.
- `get_temperature()`: Gyroskop má vestavěný senzor teploty, jejíž hodnotu ukládá do jednoho ze svých registrů. Tato hodnota může být čtená, ovšem sama o sobě nedává žádný smysl a v dokumentaci není popsán žádný postup, jak hodnotu přečtenou z registru převést na stupně Celsia. Postup byl nakonec odhalen reverzním inženýrstvím a experimentováním. Předpis pro získání teploty z hodnoty z registru je tedy $t = 36 - r$, kde t je teplota ve stupních Celsia a r hodnota z registru.
- `check_shake()`: Metoda implementuje jednoduchý detektor otřesů. Metoda porovnává aktuální hodnoty získané z gyroskopu s hodnotami získanými při posledním volání funkce a pokud v nějaké ose přesáhne rozdíl danou mez, metoda vrátí `true`. Metoda tedy musí být volána přiměřeně často.
- `detektor_otresu()`: Kostra pro doplnění během tutoriálu.

Třída `Magnetometer`

Třída ovládá magnetometr z modulu minIMU-v3.

Kromě samotných funkcí na získávání hodnoty obsahuje i dvě různé funkce pro výpočet úhlu mezi předkem Trilobota a magnetickým severem, `heading_1()` a `heading_2()`. Dvě různé funkce byly zvoleny proto, že přesnější (`heading_1()`) využívá vektorovou aritmetiku, jejíž vysvětlení a zopakování může zabrat relativně hodně času a přitom není pro zbytek tutoriálu zapotřebí a tutoriál by se zbytečně zdržoval. Proto byla vytvořená jednodušší funkce `heading_2()`, která tuto vektorovou aritmetiku nepoužívá (za cenu poněkud nižší přesnosti). Obě tyto metody v současnosti vracejí úhel pouze od 0 do 180 stupňů a tedy nejsou plně vhodné pro kompas (není možné poznat, zda se robot stáčí na východ nebo na západ).

V konstruktoru probíhá základní nastavení gyroskopu – nastavení rozsahu, frekvence nových dat a podobně. Níže jsou poté uvedeny navrhované metody:

- `get_raw_data()`: Metoda získává surová data z registrů senzoru a vrací je.
- `get_intensity()`: Nejprve získá surová data a ta poté převede na hodnoty intenzity magnetického pole ve třech osách X,Y,Z. Převod je závislý na konkrétních hodnotách citlivosti.
- `heading()`: Metoda implementující kompas. Tato metoda je zjednodušenou verzí funkce pro výpočet úhlu mezi severem a daným vektorem od výrobce modulu minIMU-v3 [32]. Funkce ovšem byla lehce upravena. Princip je následující: pomocí vektoru magnetické intenzity a vektoru dat z gyroskopu (které určují kolmý směr za předpokladu, že je robot v klidu) se určí poloha východu v rovině. Následně se z tohoto vektoru

- `heading_simple()`: Jedná se o další zjednodušení funkce `heading()`. Zanedbává se to, že vektor intenzity je prostorový a počítá se pouze úhel, který spolu svírají 2D vektory intenzity magnetického pole a předku robota[5]
- `kompas_1()`: Kostra pro doplnění složitějšího kompasu.
- `kompas_2()`: Kostra pro doplnění jednoduššího kompasu.

Třída `Accelerometer`

Tato třída ovldádá akcelerometr z modulu minIMU-v3. Kromě funkcí pro získávání surových dat a dat o akceleraci obsahuje o funkci, implementující jednoduchý detektor dopadu.

- `get_raw_data()`: Metoda získává surová data z registrů senzoru a vrací je.
- `get_acceleration()`: Ze surových dat vypočítává zrychlení ve všech třech osách a následně ho vrací.
- `impact_detector()`: Metoda má předobraz v detektorech nárazu, které se občas dají najít na balících. Pokud dojde k nárazu[6], je statická proměnná `impact` do stavu `true`, ve kterém již setrvává, až do případného resetu funkce.
- `detektor_narazu()` Kostra pro implementaci během tutoriálu.

Třída `AHRS`

Třída AHRS je jakýmsi pokusem o implementaci AHRS, tedy směrového a pozičního referenčního systému pomocí Komplementárního filtru. Jako základ pro tuto snahu byla využita bakalářská práce Julie Rakovcové [34]. Počítá se s tím, že AHRS jednou za zhruba 50 ms aktualizuje polohu Trilobota na základě dat z akcelerometru, magnetometru a gyroskopu. Pro tento přibližně stejný čas aktualizace polohy byl v hlavním souboru `trilobot.ino` vytvořen i podmíněný blok kódu, který je spouštěn pomocí přerušení, jak již bylo zmíněno.

Jelikož cílem této bakalářské práce nebylo implementovat AHRS ani žádný podobný systém, je částečně funkční kód ponechán jako vedlejší součást celého výsledku s tím, že je možné na něj libovolně navázat v budoucnu. Podrobnosti o současném stavu AHRS a jeho implementaci se dají najít v doprovodném textu k české verzi slajdů.

V konstruktoru jsou opět nastavovány hodnoty jako frekvence nových dat, rozsah a podobně (Přehled funkcí můžete vidět níže:

- `get_euler_angles()` Funkce vrátí Eulerovy úhly, tedy otočení Trilobota v osách X, Y, Z. Bohužel, tyto úhly mají tendenci již po několika vteřinách neodpovídat realitě.
- `update_euler_angles()`: Funkce aktualizuje hodnoty Eulerových úhlů.

Kód, vyjma názvů funkcí, které je třeba doplnit, je psán anglicky, komentáře jsou ovšem psané česky. Důvodem pro psaní komentářů v češtině je lepší pochopení významu kódu ze strany studentů, který se sice učí alespoň částečně programovat v angličtině (tedy používat anglické názvy proměnných), ale na druhou stranu si nemusí lámat hlavu s anglicky psanou dokumentací, což pomůže hlavně těm, jejichž angličtina není na dobré úrovni.

[5] Tento vektor vyjadřuje orientaci senzoru vůči robotovi. Jelikož je senzor minIMU namontován v ose X opačně kvůli lepší dostupnosti, hodnota tohoto vektoru je {-1,0,0}.

[6] Za náraz je považování překročení prahového zrychlení.

5.2 Přejaté části kódu

V kódu se vyskytuje několik přejatých částí a dvě přejaté knihovny[7]. Níže popíšu všechny přejaté části kódu a důvody, které k tomu vedly.

5.2.1 Knihovna `LiquidCrystal_I2C`

Tato knihovna slouží pro práci s I^2C LCD displejem. Jelikož se jedná o zdaleka nejpoužívanější knihovnu, nebyl důvod „znovu vynalézat kolo“. Knihovna je ovšem obalena wrapperem `display.h`, který velmi zjednodušuje práci s touto knihovnou.

5.2.2 Knihovna `TimerThree`

Tato knihovna byla do práce začleněna kvůli potřebě periodického přerušení.

5.2.3 Ostatní přejaté části

Část kódu v souboru `speaker.h` byla přejata z [18]. Kód ovšem prošel několika úpravami (v původním zdroji jsou například špatné frekvence některých not). Přejatá byla také implementace funkcí `beep()` a `imperial_march()`.

V souboru `magnetometer.cpp` byla funkce `heading()` inspirována stejnojmennou funkcí z [32]. Funkce byla následně zjednodušena.

5.3 Implementace slajdů

Slajdy byly implementovány dle návrhu. Ve fázi semestrálního projektu byl vytvořen pouze modul „Měříme vzdálenost“, věnující se všem třem senzorům pro měření vzdálenosti (HC-SR04, SRF-08, infračervený senzor Sharp). Zbylé slajdy byly implementovány později, kdy již byl hotový Trilobot.

V této další fázi byla sepsána kompletní verze tutoriálu pro studenty vysokých škol (nejvyšší úroveň). Tato verze obsahuje kompletní tutoriál, popisující práci se všemi hlavními periferiemi Trilobota. Výjimku tvoří reproduktor, jelikož práce s ním by byla ve skupině více studentů komplikovaná kvůli hluku a je tedy uveden pouze v přehledu periferií. Dále se tutoriál nevěnuje samostatně displeji, který slouží pro co možná nejjednodušší výpis dat a samostatný modul tutoriálu tedy nebyl nutný (ovládání probíhá pomocí dvou metod).

Verze pro střední školy také obsahuje kompletní tutoriál, je ovšem zjednodušený. Největší míra zjednodušení je vidět v odstranění nebo usnadnění některých matematických výpočtů, které pro studenty vysokých škol nejsou výraznou překážkou, ale pro středoškoláky mohou představovat zbytečné zatížení. Vysvětlování některých vzorců by také mohlo zbytečně zpomalit tutoriál, zvlášť pokud by se ve skupině vyskytovali studenti, kteří mají s matematikou problémy. Matematika by také mohla některé studenty zbytečně vyděsit.

Dále byla odstraněná i teoretická pasáž o sběrnici I^2C , jelikož studenti nutně nepotřebují vědět, jak ta do detailu funguje. Přehled funkcí ale zůstal zachován.

Verze pro základní školy je poté nejvíce zjednodušená a oproti předchozím dvěma úrovním i zkrácená. Obsahuje pouze práci se senzory vzdálenosti a s motory. Oba zachované moduly jsou přitom velmi zjednodušené. Důvodem, proč nebyly zahrnuty i ostatní moduly,

[7] Knihovny nemusely být nutně přejímány a dodávány s výsledným kódem, ale mohly být instalovány přes Arduino IDE. Tento postup se mi ovšem zdál nevhodný, jelikož by zbytečně zdržoval začátek tutoriálu za předpokladu, že by na strojích tyto knihovny nebyly přítomny.

je to, že se u studentů základních škol nepočítá s nijak hlubokou znalostí programování nebo mikrokontrolerů a pravděpodobně se bude jednat o jejich první seznámení. Tutoriál v této verzi je tedy koncipovaný tak, aby studenti mohli s naprogramovaným robotem co nejvíce interagovat.

V budoucnu by také bylo zajímavé rozšířit tuto verzi o třetí modul, umožňující studentům naprogramovat robota, který se vyhýbá překážkám (oba předchozí moduly poskytují všechny potřebné periferie i jejich naprogramování). Z časových důvodů k tomu bohužel nedošlo, ovšem v budoucnu se s tímto modulem počítá, ideálně to ve všech verzích tutoriálu.

Rozdíly v jazykových mutacích stejné úrovně jsou poté minimální a většinou vycházejí pouze z úpravy vyjádření v různých jazycích, případně drobných úpravách kvůli typografii. Význam ale zůstává nezměněn.

Kapitola 6

Testování

Během vývoje probíhala celkem dvě kola testování. Při pilotním testování bylo ověřeno, že samotná koncepce – styl psaní tutoriálu, rozdělení na slajdy a doplňkový text, využití Trilobota apod. – tvoří dohromady funkční celek.

Druhé testování mělo proběhnout po dokončení prací, ale bohužel se kvůli koronavirové epidemii uskutečnilo jen ve velmi omezené míře.

Během celého vývoje byly také jednotlivé části tutoriálu konzultovány s různými studenty, většinou s cílem zjistit názor nezaujaté osoby na danou část tutoriálu – ať již třeba samotný modul, případně na celkovou koncepci. Toto testování neprobíhalo nijak formálně a jeho výsledky proto nejsou v této práci zahrnuty. Obecně ale byly připomínky zpětně zapracovávány a tyto změny opět ověřovány na jiném studentovi.

6.1 Pilotní testování

Pilotní testování probíhalo ve chvíli, kdy byla k dispozici první kapitola tutoriálu (Měříme vzdálenost) ještě ve starém, neobjektovém provedení[1]. Úroveň tehdy byla pouze jediná, a to vysokoškolská. Testovací skupinu tvořilo 5 testerů, 2 studenti posledního ročníku střední školy a 3 vysokoškolští studenti FIT. Pilotní testování mělo především informační charakter a ověřovalo pouze „správnost koncepce“ – zda vytvářený tutoriál alespoň rámcově splňuje potřeby budoucích čtenářů – proto byl použit pouze takto omezený počet testerů. Cílem testování bylo zodpovědět následující otázky:

- Jsou slajdy a text tutoriálu srozumitelné?
- Je úroveň informací v tutoriálu dostatečná? Nechybí zde důležité informace nebo jich naopak není moc?
- Jak se pracuje s robotem?
- Je uživatel schopný tutoriál používat i samostatně (s minimální nebo žádnou interakcí s lektorem)?

Testování probíhalo s každým testerem zvlášť podle předem daného scénáře, který měl simulovat reálné použití tutoriálu. Tester měl na svém stroji k dispozici jak slajdy, tak učební text, spolu s Arduino IDE a připojeným Trilobotem.

[1] I po refaktorizaci do objektové orientace zůstala samotná logika funkcí totožná a výsledky tedy stále mají vypovídají hodnotu.

Slajdy byly následně promítané také na druhém stroji, kde jsem k nim také podával výklad. Během výkladu se mohli testeři libovolně ptát a pokud to bylo nutné, byl jsem připraven pomáhat s implementací.

Na závěr jsem studentům položil čtyři otázky popsané výše.

6.1.1 Výsledky pilotního testování

Všem se podařilo po několika desítkách minut splnit cíle tutoriálu, vysokoškoláci ho byli většinou schopní splnit i bez návodu, středoškoláci návod využívali. Učební text byl využíván méně, někteří si ho prohlíželi i po oficiálním konci testu. Dotazy se obecně neopakovaly.

Odpovědi na otázky jsou shrnuty níže:

- Všech pět testerů se shodlo na tom, že slajdy jsou obecně srozumitelné s drobnými výhradami.
- Dvěma studentům se zdálo množství informací ve slajdech i doplňkovém textu dostatečné, zbylí tři nepoužili doplňkový text, ale množství informací v tutoriálu jim přišlo dostatečné.
- Zde se všech pět studentů shodlo na tom, že je ještě na čem pracovat. Jako hlavní problém uvedli nepohyblivé senzory vzdálenosti, toho času provizorně přilepené páskou, a tím pádem nutnost natáčet celého robota.
- Tři studenti se shodli na tom, že spolu s doplňkovým textem by bylo možné tutoriál využít i při samostudiu, dva podobné použití označili za nepravděpodobné.

Při vlastním hodnocení ještě testeři zmínili, že by bylo vhodné, kdyby Trilobot uměl využít senzory vzdálenosti i jinak než jen pro samotné měření – například aby se vyhýbal překážkám. Dále se čtyři z pěti testerů shodli, že by rádi viděli alespoň názvy funkcí pro doplnění v českém jazyce, údajně kvůli lepšímu odlišení od „pevně daného“ kódu a kvůli lepší orientaci.

Z výsledků tedy vyplývá, že koncepce tutoriálu je proveditelná, i když doplňkový text není nezbytný. Na druhou stranu umožňuje minimálně části studentů využít tutoriál i bez lektora. I přes to je ovšem role lektora nezanedbatelná.

Námitky a připomínky byly následně zapracovány, jak je psáno dále. Při zpracovávání zpětné vazby jsem opravil chyby a překlepy z prvního bodu. Robot Trilobot prošel od pilotního testování velkou změnou a senzory jsou již nyní pohyblivé v horizontální rovině. Názvy funkcí jsem během refaktorizace do objektové orientace změnil na české, čímž se viditelně odlišují od ostatních funkcí, které není třeba měnit.

Poslední bod, týkající se složitějších projektů, nebyl v rámci této práce zpracováván z důvodu nízké priority. Je ovšem uveden v závěrečné části 7.1 jako jedeno z možných budoucích rozšíření. Bylo ovšem provedeno několik pokusů s tím výsledkem, že minimálně jednoduché vyhýbání se překážkám je možné implementovat poměrně jednoduše.

6.2 Závěrečné testování

Závěrečné testování mělo původně probíhat na skupině několika studentů s více funkčními Triloboty. Bohužel, kvůli koronavirové krizi toto testování nemohlo proběhnout. Po dokončení tutoriálu tedy proběhlo alespoň rychlé testování pouze na třech studentech VŠ (vizte dále).

Náhradní způsob testování probíhal přes aplikaci Skype podle následujícího scénáře:

- Tester dostal k dispozici slajdy a učební text.
- Přes sdílení obrazovky jsem předvedl konkrétní modul, přičemž jsem neukázal konečné řešení.
- Tester měl za úkol implementovat funkci na základě kostry a návodu v modulu.
- V jakékoli části mě tester mohl přerušit a zeptat se na libovolnou otázku.
- Tester mi poslal výslednou funkci a spolu jsme ji porovnali s řešením ve slajdech.
- Tento postup byl opakován vždy u několika modulů tak, aby byl každý modul vyzkoušen alespoň jednou.

Na závěr se mohli testeři k tutoriálu libovolně vyjádřit.

Z výsledků vyplynulo, že testeři jsou schopní porozumět návodu v tutoriálu velmi dobře a případné drobné chyby by jednoduše opravili.

Celkový dojem z tutoriálu byl velmi dobrý, všichni tři testeři ocenili praktické zaměření, například odkazy do dokumentace[2]. Jako rozporuplné byly hodnoceny české názvy funkcí pro doplnění. Dva testeři ocenili, že jsou odlišené funkce pro doplňování a pro běžné použití, třetí toto ovšem hodnotil jako největší slabinu tutoriálu.

Při začleňování výsledků bylo opraveno velké množství překlepů, ovšem žádné velké změny implementovány nebyly.

Testování nižších verzí tutoriálu bohužel neproběhlo. Ve středoškolské verzi ovšem proběhla vzdálená konzultace s několika studenty víceletého gymnázia se zájmem o techniku a programování, ovšem bez zkušeností s platformou Arduino, zda tutoriál neobsahuje na první pohled nesrozumitelné pasáže. Žádné takové pasáže identifikovány nebyly, ale vzhledem k malému počtu testerů a chybějícímu „ostrému“ testování jsou výsledky spíše orientační.

Pro testování nejnižší verze tutoriálu bohužel autor nemohl najít vhodné testery a celý tutoriál této verze je implementován co možná nejjednodušeji (z pohledu autora), aby bylo zaručeno, že nebude zbytečně přeceňovat schopnosti studentů.

[2]Odkazy do dokumentace samy o sobě nejsou potřebné k vyřešení tutoriálu – v návodu jsou data z dokumentace vždy přepsaná, například kdyby se studentovi nedařilo informaci najít.

Kapitola 7

Závěr

Cílem této práce bylo implementovat tutoriál pro platformy Arduino a Trilobot se zaměřením na robotiku. Prvním krokem bylo detailní seznámení s platformou Arduino a shrnutí jejich hlavních rysů. Následně byly analyzovány některé současné tutoriály jak v českém, tak anglickém jazyce a bylo provedeno jejich shrnutí.

Po obeznámení se se současným stavem byl navrhnut tutoriál, skládající se z jednotlivých modulů, kde se každý modul věnuje jednomu konkrétnímu problému, například zprovoznění senzoru, případně implementace nějaké funkcionality. V této podobě byl také tutoriál implementován. Oproti návrhu byl přidán modul se základy programování Arduina a OOP, jelikož obslužný kód robota Trilobot je psán objektově.

Tutoriál byl vypracován ve třech verzích podle předpokládané úrovně studentů. Testování, které mělo přinést poměrně důležité výsledky, ovšem proběhlo jen ve velmi omezené míře a i když jsou tedy verze tutoriálu vytvořené s co možná největší snahou správně odhadnout znalosti cílové skupiny, tento odhad je ověřen buď jen orientačně nebo vůbec.

Kromě samotného tutoriálu (výukových slajdů) vznikl pro českou mutaci i doplňkový text, jehož cílem je podrobněji hovořit o jednotlivých modulech. V úvodu se také věnuje robotice jak obecně, tak přímo na FIT VUT. Tento text je k dispozici pouze v jedné znalostní úrovni, ovšem nemělo by být těžké pro začátečníka některé složitější pasáže přeskočit.

Spolu s návrhem tutoriálu vznikal i návrh robota Trilobot a jeho kódu. Obslužný kód i samotný robot jsou v konečném důsledku samostatně funkčním celkem, který může být volně rozšiřován a díky množství funkcí mohou na základě tohoto kódu vznikat i různé projekty, jednomu z nich je věnován i odstavec v sekci 7.1.

Bohužel, kvůli situaci na jaře roku 2020 – koronavirové krizi – nebylo možné v plném rozsahu realizovat testování na skupině uživatelů. Stejně tak nebylo možné dokončit více robotů Trilobot pro reálné nasazení. Namísto toho proběhlo testování na jednotkách uživatelů dálkově formou konzultace slajdů.

7.1 Budoucí vývoj

Jak již bylo zmíněno výše, výsledek nemohl být objektivně otestován na větším počtu uživatelů při reálné zkoušce a toto tedy bude napraveno ve chvíli, kdy to podmínky dovolí, pravděpodobně během léta nebo podzimu 2020. Po tomto testování budou zapracovány případné změny podle ohlasu uživatelů. V tomto časovém horizontu budou také dokončeni další roboti Trilobot.

Během testování se také zaměřím na ověření domněnky ze sekce 4.2.3, tedy zda jsou studenti schopní spojit si objekt reálného světa a ten zapsaný v kódu.

Jako možné rozšíření celé práce by bylo možné přidat podporu pro Bluetooth modul. Tutoriál se ve své rozšířené části (doplňkový text) věnuje i komunikaci mezi více deskami, ale prakticky se toto bohužel nedá vyzkoušet, poněvadž Trilobot žádnou konektivitou nedisponuje. Podpora Bluetooth (případné jiné vhodné formy komunikace) může být lehce začleněn do současného tutoriálu jako další modul.

Dalším možným rozšířením by mohla být tvorba složitějších modulů, kdy by například cílem nebylo jen jednoduché zprovoznění jednoho konkrétního senzoru, ale například implementace systému pro vyhýbání se překážkám. Podle pokusů prováděných během implementace je možné implementovat například jednoduchý algoritmus pro hledání cesty v bludišti jen s využitím již implementovaných funkcí. tento modul by skvěle doplňoval nejnižší znalostní verzi tutoriálu.

Posledním navrhovaným vylepšením by bylo přidání baterie, která by umožnila Trilobotovi pohybovat se i bez napájecího kabelu do elektrické sítě, což se velmi hodí například při procházení bludištěm výše.

Literatura

[1] *Arduino: About Us* [online]. [cit. 2020-01-09]. Dostupné z: https://www.arduino.cc/en/Main/AboutUs.

[2] *Arduino Store* [online]. [cit. 2020-01-09]. Dostupné z: https://store.arduino.cc.

[3] *Devantech SRF08 UltraSonic Ranger* [online]. Devantech [cit. 2019-10-15]. Dostupné z: https://www.cs.york.ac.uk/micromouse/Docs/SRF08UltraSonicRanger.pdf.

[4] *Sabertooth 2x5 User's Guide* [online]. Dimension Engineering [cit. 2019-10-15]. Dostupné z: https://www.dimensionengineering.com/datasheets/Sabertooth2x5.pdf.

[5] *Sharp GP2Y0A41SK0F* [online]. Dimension Engineering [cit. 2019-10-15]. Dostupné z: https://www.pololu.com/file/0J713/GP2Y0A41SK0F.pdf.

[6] *Ultrasonic Ranging Module HC - SR04* [online]. Elec Freaks [cit. 2020-01-09]. Dostupné z: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[7] *Wire Library* [online]. Arduino.cc, prosinec 2019 [cit. 2019-10-10]. Dostupné z: https://www.arduino.cc/en/Reference/Wire.

[8] Banzi, M. *Arduino Starter Kit - Video Tutorials by Massimo Banzi* [online]. Dostupné z: https://www.youtube.com/playlist?list=PLT6rF_I5knPf2qlVFlvH47qHvqvzkknd.

[9] Banzi, M. *Getting Started with Arduino.* 1. vyd. O'Reilly Media, 2008. ISBN 978-0-596-15551-3.

[10] Banzi, M. *Send in the clones* [online]. MASSIMO BANZI, červenec 2013 [cit. 2019-11-20]. Dostupné z: https://blog.arduino.cc/2013/07/10/send-in-the-clones/.

[11] Barragán, H. *The Untold History of Arduino* [online]. Dostupné z: https://arduinohistory.github.io.

[12] Barragán, H. *Wiring: Prototyping Physical Interaction Design.* Milano, IT, 2004. Master thesis. Interaction Design Institute Ivrea. Dostupné z: https://www.fit.vut.cz/study/thesis/7848/.

[13] Chroust, M. *Ani chytré žárovky nejsou v bezpečí. Byly zneužity k napadení počítačových sítí* [online]. mobilmania.cz, únor 2020 [cit. 2020-05-01]. Dostupné z: https://www.mobilmania.cz/clanky/ani-chytre-zarovky-nejsou-v-bezpeci-byly-zneuzity-k-napadeni-pocitacovych-siti/sc-3-a-1347432/default.aspx.

[14] Eckel, T. *NewPing Library for Arduino* [online]. Arduino.cc, únor 2017 [cit. 2019-11-15]. Dostupné z: https://playground.arduino.cc/Code/NewPing/.

[15] Horáček, I. O. *HW Kitchen* [online]. [cit. 2019-11-30]. Dostupné z: https://www.hwkitchen.cz/.

[16] Horáček, O., Voda, Z. et al. *Arduino projekty* [online]. Arduino.cz, leden 2020 [cit. 2020-04-15]. Dostupné z: https://arduino.cz/category/novinky/tutorialy/arduino-projekty/.

[17] Igoe, T. *Standalone Assembly* [online]. Arduino.cc [cit. 2019-10-02]. Dostupné z: https://www.arduino.cc/en/Main/StandaloneAssembly.

[18] James, N. *Arduino Star Wars Song for Piezo* [online]. Github.com, únor 2014 [cit. 2019-10-10]. Dostupné z: https://gist.github.com/nicksort/4736535.

[19] Ježek, A. *Arduino* [online]. [cit. 2019-12-01]. Dostupné z: https://www.itnetwork.cz/hardware-pc/arduino.

[20] Ježek, A. *Lekce 7 - Arduino - Vzdálenost s ultrasonickým modulem HC-SR04* [online]. ITNetwork, říjen 2014 [cit. 2020-05-01]. Dostupné z: https://www.itnetwork.cz/hardware-pc/arduino/arduino-mereni-vzdalenosti.

[21] Ježek, A. *Lekce 15 - Arduino a I2C sběrnice* [online]. ITNetwork, únor 2016 [cit. 2020-05-01]. Dostupné z: https://www.itnetwork.cz/hardware-pc/arduino/arduino-a-I2C-sbernice.

[22] M, L. *Arduino návody* [online]. [cit. 2020-01-01]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/.

[23] M, L. *Měřič vzdálenosti ultrazvukový* [online]. Arduino návody, březen 2016 [cit. 2019-10-10]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/meric-vzdalenosti-ultrazvukovy.html.

[24] M, L. *Rotační enkodér KY-040* [online]. Arduino návody, červenec 2016 [cit. 2019-10-10]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/rotacni-enkoder-ky-040.html.

[25] M, L. *Arduino Bluetooth 4.0 BLE modul HM-10* [online]. Arduino návody, srpen 2017 [cit. 2020-05-01]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/arduino-bluetooth-4.0-ble-modul-hm-10.html/.

[26] M, L. *Arduino Bluetooth 4.0 BLE modul HM-10* [online]. Arduino návody, únor 2017 [cit. 2020-05-01]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/esp8266-vyvojova-deska-wemos-d1.html.

[27] M, L. *Arduino Bluetooth modul HC-05* [online]. Arduino návody, květen 2017 [cit. 2020-05-01]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/arduino-bluetooth-modul-hc-05.html.

[28] M, L. *Ultrazvukový měřič vzdálenosti HY-SRF05* [online]. Arduino návody, únor 2018 [cit. 2019-10-10]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/ultrazvukovy-meric-vzdalenosti-hy-srf05.html.

[29] Malý, M. *Senzory Martina Malého: Průvodce světem Arduina* [online]. [cit. 2019-12-01]. Dostupné z: https://www.lupa.cz/clanky/senzory-martina-maleho-pruvodce-svetem-arduina/.

[30] Malý, M. *Hradla volty jednočipy: Úvod do bastlení.* 1. vyd. CZ.NIC, 2017. ISBN 978-80-88168-23-2.

[31] MLubo, *Gyroskop, akcelerometr a magnetometr MPU-9250* [online]. Arduino návody, prosinec 2017 [cit. 2019-10-10]. Dostupné z: https://navody.arduino-shop.cz/navody-k-produktum/gyroskop-akcelerometr-a-magnetometr-mpu-9250.html.

[32] Mulligan, R. et al. *LSM303.h* [online]. Github.com, srpen 2016 [cit. 2020-15-01]. Dostupné z: https://github.com/pololu/lsm303-arduino/blob/master/LSM303.h.

[33] Polesný, D. *Živě.cz* [online]. [cit. 2019-11-30]. Dostupné z: https://www.zive.cz.

[34] Rakovcová, J. *Návrh a implementace AHRS systému založeného na MEMS senzorech.* Plzeň, CZ, 2016. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Dostupné z: https://dspace5.zcu.cz/bitstream/11025/23776/1/BP_Rakovcova.pdf.

[35] Rickman, J. *LiquidCrystal I2C* [online]. Github.com, duben 2019 [cit. 2020-05-01]. Dostupné z: https://github.com/johnrickman/LiquidCrystal_I2C.

[36] Robotics, A. *Trilobot Research Robot* [online]. Arrick Robotics, 1998 [cit. 2020-15-01]. Dostupné z: https://www.arrickrobotics.com/trilobot/.

[37] Rodríguez Corral, J., Amaya Rodriguez, C., Civit, A., Morgado Estevez, A., Perez Peña, F. et al. Application of Robot Programming to the Teaching of Object-Oriented Computer Languages. *International Journal of Engineering Education.* Červenec 2016, sv. 32, s. 1823–1832.

[38] scijoy. *Exploring Encoders - Step Trackers for Motors* [online]. Arduino.cc [cit. 2019-10-02]. Dostupné z: https://create.arduino.cc/projecthub/scijoy/exploring-encoders-step-trackers-for-motors-427a99?ref=search&ref_id=encoder&offset=0.

[39] Stoffregen, P. *TimerThree* [online]. Github.com, prosinec 2019 [cit. 2020-15-01]. Dostupné z: https://github.com/PaulStoffregen/TimerThree.

[40] Taibo, N. *HC-SR04 Distance Sensor Service* [online]. Arduino.cc, březen 2016 [cit. 2019-10-10]. Dostupné z: https://create.arduino.cc/projecthub/now/hc-sr04-distance-sensor-service-9179e1?ref=search&ref_id=hc-sr04&offset=0.

[41] Voda, Z. et al. *Průvodce světem Arduina.* 2. vyd. Martin Stříž, 2017. ISBN 978-80-87106-93-8.

[42] Čápka, D. et al. *Největší česká IT akademie* [online]. [cit. 2019-11-30]. Dostupné z: https://www.itnetwork.cz.

[43] Čížek, J. *Pojďme programovat elektroniku* [online]. Dostupné z: https://www.zive.cz/pojdme-programovat-elektroniku/sc-695/default.aspx.

[44] Čížek, J. *Pojďme programovat elektroniku: Vyzkoušíme IR, ovládneme světýlko přes Bluetooth a vyšleme zprávu na sto metrů* [online]. Živě.cz, září 2016 [cit. 2019-10-10]. Dostupné z: https://www.zive.cz/clanky/pojdme-programovat-elektroniku-vyzkousime-ir-ovladneme-svetylko-pres-bluetooth-a-vysleme-zpravu-na-sto-metru/sc-3-a-184279/default.aspx.

[45] Čížek, J. *Pojďme programovat elektroniku: Vyzkoušíme ultrazvukový dálkoměr, detektor pohybu, deště a další kouzla* [online]. Živě.cz, srpen 2016 [cit. 2019-10-10]. Dostupné z: https://www.zive.cz/clanky/pojdme-programovat-elektroniku-vyzkousime-ultrazvukovy-dalkomer-detektor-pohybu-deste-a-dalsi-kouzla/ultrazvukovy-dalkomer-hc-sr04/sc-3-a-183990-ch-103776/default.aspx#articleStart.

[46] Čížek, J. *Pojďme programovat elektroniku: Vyzkoušíme elektronický papír, který proslavil čtečku Kindle* [online]. Živě.cz, říjen 2017 [cit. 2019-10-10]. Dostupné z: https://www.zive.cz/clanky/pojdme-programovat-elektroniku-vyzkousime-elektronicky-papir-ktery-proslavil-ctecku-kindle/sc-3-a-190206/default.aspx.

[47] Čížek, J. *Pojďme programovat elektroniku: Rádiový čip, který má skoro každá bezdrátová myš* [online]. živě.cz, prosinec 2019 [cit. 2020-05-01]. Dostupné z: https://www.zive.cz/clanky/pojdme-programovat-elektroniku-radiovy-cip-ktery-ma-skoro-kazda-bezdratova-mys/sc-3-a-201436/default.aspx.