

Univerzita Karlova  
Pedagogická fakulta

DIPLOMOVÁ PRÁCE

2024

Bc. Oliver Erbes

Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

## DIPLOMOVÁ PRÁCE

Možnosti edukační robotiky ve výuce STEM

Possibilities of Educational Robotics in STEM Education

Bc. Oliver Erbes

Vedoucí práce: PhDr. Daniel Tocháček

Studijní program: Učitelství informačních a komunikačních technologií pro 2. stupeň  
základní školy a střední školy (N0114A300094)

Studijní obor: N IKT-M 20 (0114TA300094, 0114TA300095)

Prohlašuji, že jsem diplomovou práci na téma Možnosti edukační robotiky ve výuce STEM vypracoval pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále prohlašuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

V Praze 20. 6. 2024

.....

podpis

Rád bych poděkoval vedoucímu mé práce PhDr. Danielovi Tocháčkovi za podstatné rady, příjemnou spolupráci a předané znalosti k této práci. Dále bych chtěl poděkovat ZŠ Na Lukách za poskytnutí prostor k vyučování a žákům za práci. V neposlední řadě bych chtěl poděkovat rodině a kamarádům za podporu a pochopení.

## **ABSTRAKT**

Diplomová práce se zaměřuje na využití edukační robotiky ve výuce STEM předmětů na základní škole s cílem prohloubit vztahy mezi informatikou, matematikou a fyzikou. Cílem je navrhnout a ověřit náměty úloh postavené na robotickém projektu využívající mikrokontrolery Arduino. Úlohy zahrnují 3D modelování, 3D tisk, konstrukci robotických systémů a jejich programování. Tyto náměty budou testovány v praxi prostřednictvím případové studie, realizované ve výuce s žáky devátého ročníku základní školy. Empirická část práce se detailně zabývá realizací případové studie, včetně definování výzkumných otázek, formulace hypotéz, stanovení vzdělávacích cílů a analýzy výsledků žáků. Sběr a analýza dat slouží k objektivnímu hodnocení efektivity a přínosů navržených metodických materiálů pro praxi výuky. Výsledkem této práce je soubor ověřených námětů a metodických materiálů, které podporují rozvoj praktických dovedností žáků a integrují moderní technologie do výuky STEM předmětů na základních školách. Tento přístup přispívá k posílení aplikovaného učení a připravuje žáky na efektivní uplatnění v novodobé společnosti.

## **KLÍČOVÁ SLOVA**

Edukační robotika, STEM, Arduino, robotika, konstruktivismus, 2. stupeň ZŠ

## **ABSTRACT**

This thesis focuses on the use of educational robotics in the teaching of STEM subjects in primary school in order to deepen the relationships between computer science, mathematics and physics. The aim is to design and validate task themes based on a robotics project using Arduino microcontrollers. The tasks include 3D modelling, 3D printing, construction of robotic systems and their programming. These themes will be tested in practice through a case study implemented in a classroom with ninth grade elementary school students. The empirical part of the thesis deals in detail with the implementation of the case study, including the definition of the research questions, the formulation of hypotheses, the setting of learning objectives and the analysis of the pupils' results. Collection and analysis of data used to objectively evaluate the effectiveness and benefits of the proposed methodological materials for teaching practice. The result of this work is a set of validated themes and methodological materials that support the development of students' practical skills and integrate modern technologies into the teaching STEM subjects in primary schools. This approach contributes to enhance applied learning and prepares pupils for effective employment in modern society.

## **KEYWORDS**

Educational robotics, STEM, Arduino, robotics, constructivism, elementary school

## Obsah

1	Robotika .....	9
1.1	Pohon a manipulace.....	13
1.2	Edukační robotika a konstruktivismus .....	14
1.2.1	Metodické přístupy .....	16
2	Projekt pomocí konceptu STEM .....	20
2.1	STEM .....	20
2.1.1	The house of STEM.....	24
2.2	3D modelování .....	27
2.2.1	Primitiva .....	28
2.2.2	Transformace .....	29
2.3	Mikrokontrolery Arduino .....	33
2.3.1	Popis desky Arduino UNO .....	34
2.3.2	Dosavadní možnosti využití Arduino na 2. stupni ZŠ.....	36
2.3.3	Elektrotechnické součástky .....	38
2.3.4	Pohon – krokový motor .....	39
2.3.5	Senzory .....	42
2.4	Programování mikrokontroleru Arduino .....	43
2.4.1	Struktura programu.....	44
2.4.2	Funkce .....	45
3	Empirická část .....	47
3.1	Výzkumná otázka .....	48
3.2	Robotické auto.....	48
3.2.1	Navržení 3D modelu.....	49
3.2.2	3D tisk, stavba a zapojení .....	52

3.2.3	Programování .....	55
3.3	Návrh námětů .....	59
3.4	Charakteristika účastníků případové studie.....	72
3.5	Sběr dat pro případovou studii a jejich analýza.....	73
3.5.1	Pozorování .....	73
3.5.2	Rozhovory .....	82
3.5.3	Test .....	85
3.5.4	Analýza dat z testů.....	87
3.5.5	Dotazník .....	91



# Úvod

V současné době si můžeme povšimnout zájmu o inovace ve vzdělávání, které reagují na stále postupující technologický pokrok a měnící se požadavky na dovednosti žáků. V kontextu těchto změn se práce zaměří na začlenění edukační robotiky do výuky STEM předmětů (Science, Technology, Engineering, Mathematics) na základních školách. Tento přístup rozvíjí jak žákovské teoretické znalosti, tak i praktické dovednosti ve všech oblastech. Díky tomu žáci získají možnost snáze uchopit učivo pro reálné aplikace v životě.

Diplomová práce se zaměřuje na konkrétní možnosti využití edukační robotiky ve výuce STEM předmětů na druhém stupni základních škol, konkrétně pro vzdělávací oblasti matematika, fyzika a informatika. Hlavním cílem je navrhnout a ověřit náměty pro sestavení robotického systému s využitím mikrokontroleru Arduino. Arduino je otevřená platforma pro tvorbu elektronických projektů, která je díky své jednoduchosti a dostupnosti ideální pro edukační potřeby. Ačkoli je využití Arduina často spojováno s projekty na středních školách, tato práce bude zkoumat, zda je možné efektivně integrovat tuto technologii i na úrovni základní školy. Práce se skládá z části teoretické a empirické.

Teoretická část práce se zabývá vymezením základních pojmů souvisejících s robotikou, edukační robotikou, konceptem STEM, popisem platformy Arduino a možnostmi jejího programování. Dále se práce zaměřuje na význam konstruktivistických přístupů ve výuce a přehled dosavadních možností využití Arduino v STEM metodě na druhém stupni základních škol.

Empirická část zahrnuje návrh konkrétních námětů pro zapojení Arduina do výuky předmětů STEM, popis robotického projektu a realizace projektu s žáky. Výzkum bude probíhat formou případové studie, která ověří navržené náměty v praxi. Součástí je položení výzkumné otázky a formulace hypotéz. Následně budou navržené úlohy implementovány do výuky s žáky devátého ročníku základní školy. V rámci případové studie bude proveden sběr a analýza dat, které umožní zhodnotit efektivitu a přínos navržených aktivit a odpovědět na výzkumnou otázku.

# 1 Robotika

Robotika představuje technologii s významným sociálním a ekonomickým potenciálem. Zkoumání, jak roste sofistikovanost robotů a jejich rozsáhlé nasazení od domácností přes nemocnice, veřejné prostory až po válečné technologie na bojišti, vyžaduje přehodnocení široké škály filozofických i politických otázek. Ano, robotika se nachází v období rozvoje a růstu, avšak stále ještě není na úrovni sci-fi představ, kde jsou roboti tak vyspělí, že téměř nelze rozeznat mezi nimi a lidmi, jak je popsáno v literárním díle R. U. R. od Karla Čapka.<sup>1</sup> V tomto díle je poprvé použito slovo "robot". To můžeme považovat za českou hrdost, protože mělo vliv na popularizaci tohoto slova, které si převzal celý svět. V současné době se věnujeme nejen zdokonalování schopností robotů, ale také řešení technických, bezpečnostních, etických a právních otázek spojených s jejich použitím a integrací do společnosti. Robotika dává a zároveň bere, umožňuje převzít nepříjemné, obtížné nebo nudné úkoly. Tato automatizace v průmyslu může mít za důsledek ztrátu pracovních míst nebo změnu pracovních rolí pro lidské pracovníky. Lidská práce je velmi drahá, tudíž pro firmy je to velikou příležitostí. Okolo nás doma se setkáváme s robotickými vysavači, po světě se objevují první taxíky bez řidičů, fastfoodové restaurace, kde jsou zákazníci obsluhováni roboty. Ti jsou schopni vytvářet objednávku podle QR kódů, které jsou rozesety po celé kuchyni, tím, že už je robotika vyzrálá věda. Tento trend může působit jako utopie nebo jako vizionářská představa budoucnosti.

Co nazýváme robotem? Můžeme si představit objekt schopný reagovat na podněty pomocí senzoru, nebo vstupního mechanismu, který je ovládán nějakým algoritmem, který řídí reakce na zaznamenaná data a nabývá schopnosti reagovat tak, že ovlivňuje nebo je alespoň znatelný světu vně kolem robota. Roboti mohou mít velmi úzký rozsah funkce, nebo mohou mít mnoho funkcí, to platí i pro rozsah pohybu a rozhodování.<sup>2</sup>

---

1/ ČAPEK, Karel. R.U.R.: Rossum's universal robots. V nakladatelství Artur vydání čtvrté. D (Artur). Praha: Artur, 2021. ISBN 978-80-7483-150-8.

2/ FROOMKIN, Michael. Robot Law. University of Miami School of Law, 2016. ISBN 978-1-78347-672-5.

Matičkova definice pojmu **robot**<sup>3</sup>: „Robot je automaticky nebo počítačem řízený integrovaný systém, schopný autonomní cílové orientované interakce a přirozeným prostředím podle instrukcí od člověka. Tato interakce spočívá (a) ve vnímání a rozpoznávání tohoto prostředí a (b) v manipulování s předměty, popř. pohybování se v tomto prostředí.“

Roboty můžeme dělit podle toho, jak se postupně vyvíjeli, konkrétně na dělení manipulátoru a robotu z hlediska ovládání a programování. Dělíme je na manipulátory, které slouží k mechanické manipulaci bez jakékoli formy inteligence. Následují synchronní manipulátory (teleoperátory), které jsou ovládané pracovníkem a poskytují posílení jeho síly. Poté máme programovatelné roboty, kteří mohou mít pevně stanovený nebo měnitelný program a kognitivní roboty, ti jsou vybaveni vnímáním a racionálním myšlením.<sup>4</sup>

Věda zabývající se touto problematikou a tvorbou robotů je **robotika**<sup>5</sup>, ta se poprvé objevuje v povídce „Liar!“ od Isaaca Asimova, který ji definuje jako „vědeckou disciplínu zabývající se studiem, návrhem, vývojem a použitím robotů“.

Existují různé přístupy k robotice, zahrnující:<sup>6</sup>

1. Teoretickou robotiku: zkoumá základní principy a možnosti robotiky z různých perspektiv, jako je biologie, psychologie, etologie, matematika a fyzika. Zaměřuje se na porozumění základních principů a omezení.
2. Experimentální robotiku: testuje a ověřuje tyto teoretické principy v praxi, staví prototypy a experimentální zařízení, často nazývané "hračky". Zahrnuje oblasti jako kybernetika, umělá inteligence a různé inženýrské disciplíny.

---

3/ MATIČKA, Robert. KONSTRUKCE PRŮMYSLŮVÝCH ROBOTŮ A MANIPULÁTORŮ. ČVUT, 1995. ISBN 80-01-01291-3. s. 8.

4/ MATIČKA, Robert. KONSTRUKCE PRŮMYSLŮVÝCH ROBOTŮ A MANIPULÁTORŮ. ČVUT, 1995. ISBN 80-01-01291-3. s. 6.

5/ ASIMOV, Isaac. I, robot. Robot series. New York: Bantam Books, 2008. ISBN 978-0-553-38256-3.

6/ HLAVÁČ, Václav. Úvod do robotiky. Online. 2024. Dostupné z: <https://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/01UvodRobotika.pdf>. [cit. 2024-04-30].

3. Průmyslovou robotiku: zabývá se návrhem, vývojem a využitím robotů ve výrobním prostředí. Zahrnuje teorii a technologie řízení, elektroniku, strojírenství, automatizaci a organizaci výroby, stejně jako znalosti specifických oblastí využití robotů v průmyslu.

4. Aplikovanou robotiku: zaměřuje se na návrh a vývoj inteligentních strojů pro různé účely, nejen v průmyslu. Například stroje pro kontrolu kvality ve výrobě, mobilní roboty s autonomní navigací a další specifické aplikace, které využívají pokročilé technologie robotiky.

Zmiňovaná povídka byla již třetí autorovou povídkou o robotech. Tato povídka byla následně (r. 1950) zařazena do autorových sbírek „I, robot“. V této sbírce Asimov definuje **3 zákony robotiky** <sup>7</sup>:

*„1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.*

*2. Robot musí uposlechnout příkazů člověka, kromě případů, když jsou tyto příkazy v rozporu s prvním zákonem.*

*3. Robot musí chránit sám sebe před zničením, kromě případů, kdy je tato ochrana v rozporu s prvním nebo druhým zákonem.“*

Asimov dodatečně (r. 1985) v knize „Robots and Empire“ přidal nultý zákon <sup>8</sup>, který u fanoušků vyvolal smíšené reakce svojí problematikou. Zákon totiž robotům dovoluje ochraňovat celé lidstvo na úkor jedinců. Zní následovně:

*„0) Robot nesmí ublížit lidstvu nebo svou nečinností dopustit, aby mu bylo ublíženo.“*

Musíme dodat, že Asimov tyto zákony používal spíše jako zápletkové prvky v souvislosti s roboty než jako návrhové parametry pro reálnou verzi zákonů pro dnešní roboty. V blízké budoucnosti budou naše životy transformovány sofistikovanějšími roboty. Ve srovnání se současnými roboty budou mít mnohem vyšší úroveň konektivity, v tom smyslu, že stroje mohou nezávisle přijímat a přenášet informace, autonomie, jakožto nezávislou schopnost

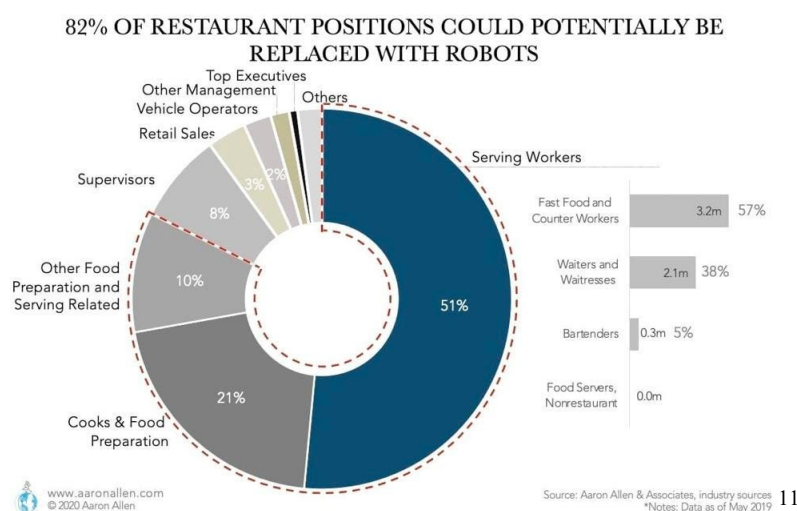
---

7/ Tamtéž

8/ ASIMOV, Isaac. Robots and Empire. Argo, 2012. ISBN 978-80-257-0720-3.

reagovat na vnější vstupy a reagovat s okolím a inteligencí. Když mluvíme o inteligenci stojí za zmínku umělá inteligence, která jde ruku v ruce s robotikou.<sup>9</sup>

AI a robotika společně představují klíčové technologické oblasti, které mohou revolučně změnit průmyslové procesy. Integrace AI do robotiky umožňuje vznik „inteligentních robotů“, kteří jsou schopni provádět sofistikované úkoly s využitím pokročilé analýzy dat. Tato kombinace technologií přináší významné možnosti automatizace a optimalizace průmyslových procesů.<sup>10</sup>



Obr. 1 – Rozložení potenciálně nahraditelných pozic roboty v restauracích

Podle společnosti Aaron Allen & Associates by až 82 % pracovních pozic v restauracích mohlo být potenciálně převzato roboty, z toho až 31 % by mohlo být věnováno přípravě jídla. Tento trend směrem k automatizaci by mohl přinést významné úspory pro fastfoodové provozy, které jsou odhadovány na více než 12 miliard dolarů ročně, jak uvádí tato skupina.<sup>12</sup>

9/ FROMKIN, Michael. Robot Law. University of Miami School of Law, 2016. ISBN 978-1-78347-672-5.

10/ URBANOVÁ, Libuše. Umělá inteligence jako součást robotiky a její využití v dnešním světě. Online. Alfun. 2023. Dostupné z: <https://www.alfun.cz/umela-inteligence-jako-soucast-robotiky-a-jeji-vyuziti-v-dnesnim-svete/>. [cit. 2024-04-29].

11/ EWING-CHOW, Daphne. Here Are Five Global Restaurants Staffed By Robot Chefs. Online. Forbes. 2024. Dostupné z: <https://www.forbes.com/sites/daphneewingchow/2024/03/31/here-are-five-global-restaurants-staffed-by-robot-chefs/>. [cit. 2024-04-30].

12/ Tamtéž

## 1.1 Pohon a manipulace

K manipulaci, pohybu, vnímání prostředí a zpětné reakci robot potřebuje pohon a senzory. Ty v následující podkapitole rozebereme. Pohon robota nebo manipulátoru má za úkol přeměňovat vstupní energii na mechanický pohyb. Tuto transformaci zajišťuje motor, spolu s blokem pro řízení energie do motoru a spojovacím blokem, který ovlivňuje přenos pohybu z motoru na pohybovou jednotku. Pohyb z výstupu motoru se následně přenáší na pohybovou jednotku přímo nebo prostřednictvím transformačního bloku. Obecně klademe na pohon robota několik požadavků. Patří sem plynulý rozběh a brzdění, vysoká přesnost polohování pro použití v průmyslových nebo zdravotnických aplikacích a minimalizace hmotnosti a rozměrů. Pro různá odvětví mohou být tyto obecné požadavky nedostatečné a musí být upraveny podle konkrétních podmínek aplikace. Například pro dosažení nejkratšího pracovního cyklu je klíčová rychlost pohonu. V nemocničním prostředí je důležitá rovnoměrnost pohybu, s ohledem na oblast operace. Maximální rychlost manipulačních pohybů se může lišit, ale v některých případech může dosahovat až 5 metrů za sekundu, avšak v prostředí s lidskou přítomností nebo dalšími stroji musí být maximální rychlost omezena z důvodu bezpečnosti.<sup>13</sup>

### Dělení pohonů

Podle druhu energie přiváděné na vstup motoru můžeme rozlišovat strukturu motorů na elektrické, tekutinové a kombinované. Elektrické pohony využívají elektromotory. Tekutinový pohon zahrnuje hydraulické nebo pneumatické pohony. Kombinované pohony mohou kombinovat elektrické a tekutinové prvky buď v rámci jedné pohybové jednotky, nebo v celém robotu. To zahrnuje spojení elektromotoru s hydromotorem nebo použití elektromotoru a tekutinového motoru v různých částech robotu. Z hlediska charakteru pohybu na výstupu motoru můžeme motory dělit na:

- s otáčivým pohybem (rotační motory).
- s přímočarým pohybem (přímočaré).

---

13/ MATIČKA, Robert. KONSTRUKCE PRŮMYSLVÝCH ROBOTŮ A MANIPULÁTORŮ. ČVUT, 1995. ISBN 80-01-01291-3. s. 36.

Podle průběhu výstupního pohybu v závislosti na čase se motory dělí na:

- se spojitým pohybem.
- s krokovým pohybem (krokové motory).
- s hybridním (možnost se spojitým nebo krokovým) pohybem.<sup>14</sup>

Přímočaré motory pracují s omezeným rozsahem a jsou tedy použitelné jen v případě přetržitých vratných pohybů. Totéž platí pro rotační motory s omezeným rozsahem natočení.

Motory můžeme dělit na stejnosměrné a střídavé podle typu využívaného proudu. Stejnosměrné motory se v elektromobilitě využívaly v minulosti. Dnes převládají střídavé, které jsou méně konstrukčně složitější a více efektivní. Střídavé motory jsou dále synchronní a asynchronní. Označení synchronní/asynchronní odkazuje na základní princip fungování motorů. U synchronních motorů se rotor otáčí synchronně s magnetickým polem statoru, které je generováno vinutím. To znamená, že rychlost otáčení rotoru je přesně synchronizována s frekvencí střídavého proudu, který napájí stator. Na druhou stranu u asynchronních motorů se rotor otáčí pomaleji než magnetické pole statoru. Tento rozdíl otáček vytváří indukci napětí v rotoru, čímž se vytváří vlastní magnetické pole rotoru. Právě proto se asynchronním motorům říká také motory indukční.<sup>15</sup>

## 1.2 Edukační robotika a konstruktivismus

Robotika má uplatnění i ve vzdělávání samozřejmě v technických, ale i v rámci vzdělávání v humanitních a přírodovědných předmětech na základních a středních školách. Robotická zařízení mohou být využita jako prostředky k rozvoji analytického myšlení, kreativity a k lepšímu porozumění různým konceptům. Ve výuce mohou sloužit jako nástroj pro

---

14/ MATIČKA, Robert. KONSTRUKCE PRŮMYSLOVÝCH ROBOTŮ A MANIPULÁTORŮ. ČVUT, 1995. ISBN 80-01-01291-3. s. 39.

15/ Elektromotory – Synchronní vládne. Online. Automobil. 2022. Dostupné z: [https://www.automobilrevue.cz/rubriky/presunuto-na-trucker-cz-truck-bus/predstavujeme/elektromotory-synchronni-vladne\\_48530.html](https://www.automobilrevue.cz/rubriky/presunuto-na-trucker-cz-truck-bus/predstavujeme/elektromotory-synchronni-vladne_48530.html). [cit. 2024-04-28].

podněcování diskuse, zkoumání jednoho tématu nebo jako prostředek pro propojení různých témat a oborů s interdisciplinárním přístupem.<sup>16</sup>

Tocháček a Lapeš definují **edukační robotiku**<sup>17</sup> jako „*specifické odvětví robotiky úzce propojené s pedagogikou využívající robotů (robotických aktivit, robotických projektů atd.) jako prostředků plnění vzdělávacích cílů*“. Edukační robotika může být využita k výuce informatiky, konkrétně programování, algoritmizaci, obecně k rozvoji inforatického myšlení, ale současně v předmětech přírodovědných či humanitních. Tím se dostáváme k možnému interdisciplinárnímu přesahu, který by mohl vést k formě STEM výuky.

Konstruktivismus je hojně pedagogicky zmiňovanou a pokud možno využívanou vzdělávací teorií, na kterou se v dnešní době klade veliký důraz přes různé vzdělávací oblasti, jako je informatika, chemie, matematika, ale i humanitní vědy.<sup>18</sup> Základní myšlenkou je nabourání vztahu mezi učitelem a žákem, kde učitel v porovnání s frontální výukou (transmisivním vyučováním) jde do pozadí a odpovědnost na získání znalostí se přesouvá na žáka. Učitel jednoduše nepředává žákovi hotové informace, ale stává se spíše průvodcem, který se žáka snaží dovést k nové znalosti žakovským bádáním.<sup>19</sup> Cílem je konstruovat poznatky aktivně učícím se žákem, učení nemá být pasivní činností, proto název konstruktivismus. Žák buduje své poznatky a dovednosti postupně na základě informací a zkušeností, které postupně získává v průběhu svého života. Učení může probíhat jak individuálně, tak i skupinově, což právě spolupráce ve skupině může s různými myšlenkovými postupy jedinců vést k jednotlivým krokům k získání znalosti s vyšší pravděpodobností. Pokud bychom se bavili o časové náročnosti, je zřejmé, že bude vyšší oproti transmisivnímu vyučování. Přesto si myslím, že ve výuce by měla převyšovat kvalita probraného učiva nad kvantitou. Při tomto

---

16/ TOCHÁČEK, Daniel a LAPEŠ, Jakub. Edukační robotika. Praha: Pedagogická fakulta Univerzity Karlovy, 2012. ISBN 978-80-7290-577-5. s. 21.

17/ Tamtéž

18/ TOCHÁČEK, Daniel a LAPEŠ, Jakub. Edukační robotika. Praha: Pedagogická fakulta Univerzity Karlovy, 2012. ISBN 978-80-7290-577-5. s. 22.

19/ KOŠÁTKOVÁ, Markéta. Konstruktivismus. Online. MUNI. 2023. Dostupné z: [https://is.muni.cz/do/rect/el/estud/pdf/ps23/pvc/web/pages/03\\_01\\_05\\_konstruktivismus.html](https://is.muni.cz/do/rect/el/estud/pdf/ps23/pvc/web/pages/03_01_05_konstruktivismus.html). [cit. 2024-04-29].



bádání a objevování dochází k prohlubování pochopení jiných konceptů (např. důkazy vět v matematice, při kterých využijete jiné znalosti k dokázání nové). Každý jedinec je jiný, jinak přemýšlí a vytváří si vlastní způsoby uchopení informací a propojuje je s existujícími znalostmi. Tímto způsobem můžeme docílit, že minimalizujeme formální znalosti (bez pochopení).<sup>20</sup> Některá témata ve vzdělávání jsou vhodnější k vyučování v klasickém instruktivním stylu, který je charakterizován řízením práce studenta podle instrukcí nebo vzoru. Konstruktivismus se nemusí hodit pro každého a pro všechna témata, zejména pro ty, kteří jsou zvyklí na tradiční transmisivní přístup.

### 1.2.1 Metodické přístupy

Typickými metodami pro konstruktivisticky založenou výuku edukační robotiky je zapojení žáků do projektově orientované výuky. To může zahrnovat například vytvoření robota, který je schopný pohybu a základní inteligence (např. nenaráží do překážek). Žáci musí navrhnout, sestavit, zapojit elektrické obvody a naprogramovat robota, jak je popsáno v této diplomové práci. Další variantou je problémově orientovaná výuka, která patří mezi heuristické metody zaměřené na řešení problémů. Tato metoda staví žáky před konkrétní problémovou situací nebo úkol, kde se stávají badateli a rozvíjejí své kreativní myšlení. Učitel v tomto procesu funguje jako partner. Problémové úlohy mohou být uzavřené nebo otevřené. Příkladem může být naprogramování robota, který třídí různobarevné kostky nebo projíždí bludištěm. Didaktické hry, které zahrnují prvky gamifikace, jsou rovněž součástí edukační robotiky. Mohou obsahovat například závod po dráze na čas, kde žáci hledají nejrychlejší průchod dráhou. Je důležité pracovat i s žákovskými prekoncepty, které mohou být pro daný úkol nedostatečné. V takovém případě můžeme zvolit spíše transmisivní výuku, například přednášku k vysvětlení určité problematiky, a poté se vrátit ke konstruktivistické výuce.<sup>21</sup>

Zmiňované prekoncepty jsou základním kamenem pro konstruktivistickou výuku. Učení dítěte začíná již v předškolním věku, kdy si utváří své vlastní představy o sobě, světě

---

20/ SERAFÍN, Čestmír. CELOŽIVOTNÍ UČENÍ A KONSTRUKTIVISMUS. Online.

LifeLongLearningMendelu. 2011. Dostupné

z: [https://lifelonglearning.mendelu.cz/media/pdf/LLL\\_20110102007.pdf](https://lifelonglearning.mendelu.cz/media/pdf/LLL_20110102007.pdf). [cit. 2024-04-29].

21/ Výukové metody aktivizující. Online. RVP. 2012. Dostupné z:

<https://clanky.rvp.cz/clanek/c/s/15017/VYUKOVE-METODY-AKTIVIZUJICI.html>. [cit. 2024-06-10].

a ostatních lidech. Tyto představy jsou formovány skrze postupně nabyté zkušenosti a předpoklady. Podnětnou částí je konfrontace žáka při konstruktivistické výuce s jeho prekoncepty. Cílem je začlenění nových poznatků do mentální struktury, podobně jako při skládání do skladiště. Učitel má za úkol facilitovat tento proces, poskytovat podnětné informace, reflektovat aktuální pochopení žáka a být připraven vést s ním dialog.<sup>22</sup>

Kooperativní výuka je v kombinaci s těmito metodami dle mého klíčem k úspěchu. Skupina osob má vyšší pravděpodobnost úspěchu než jedna. Podstatnou částí je kooperace a spoléhání na sebe, svoji skupinu, kde má každý svoji roli. Každý ve skupině je teoreticky lepší v jiné aktivitě a tím se mohou doplňovat. Výhodou je, že žák při této koncepci naplňuje všechny kompetence dle RVP. Kooperativní výuka<sup>23</sup> splňuje 5 těchto principů:

1. *Třída je rozdělena do malých různorodých skupin (obvykle 3-5 žáků) tak, aby žáci mohli komunikovat tváří v tvář.*
2. *Základní vztah mezi žáky ve skupině při plnění úkolu je kooperace, tj. úspěch jednotlivce v dosažení cíle úkolu je vázán na úspěch dalších členů skupiny.*
3. *Součástí učení je využívání a formování sociálních dovedností pro práci ve skupině.*
4. *Sleduje se také to, jak jednotlivci přispívají ke skupinovému úsilí, co a jak se jednotlivý žák učil ve skupinové práci.*
5. *Učení ve skupinách se podrobuje reflexi – nejen v tom, co se děti učily, ale také jak fungovaly sociální vztahy a dovednosti v učení.*

Otcem **konstruktivismu**<sup>24</sup> je psycholog Jean Piaget a podle něho „vzdělávání není správné chápat jako proces přenosu informací od učitele k žákovi nebo studentovi, ale jako proces aktivního vlastního konstruování znalostí založený na zkušenostech daných skutečným životem, které se nabalují na dříve získané znalosti a zkušenosti“.

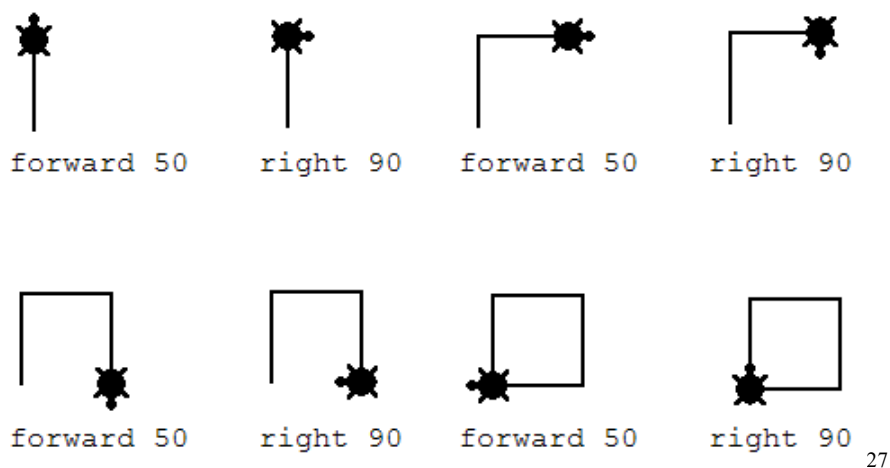
---

22/ KOŠÁTKOVÁ, Markéta. Konstruktivismus. Online. MUNI. 2023. Dostupné z: [https://is.muni.cz/do/rect/el/estud/pedf/ps23/pvc/web/pages/03\\_01\\_05\\_konstruktivismus.html](https://is.muni.cz/do/rect/el/estud/pedf/ps23/pvc/web/pages/03_01_05_konstruktivismus.html). [cit. 2024-04-29].

23/ Kooperativní výuka. Online. RVP. 2011. Dostupné z: [https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD\\_lexikon/K/Kooperativn%C3%AD\\_v%C3%BDuka](https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD_lexikon/K/Kooperativn%C3%AD_v%C3%BDuka). [cit. 2024-06-10].

24/ PIAGET, Jean. The Principles of Genetic Epistemology. Psychology Press, 1997. ISBN 978-0710072962.

Tyto myšlenky doplňuje Seymour Papert teorii konstrukcionismu<sup>25</sup> tak, že nejúčinnějším způsobem konstrukce poznatků je tvorba osobně zajímavých produktů, projektů. Tím, že v našem případě budou žáci konstruovat robota od nuly, mohou si s ním vytvořit vztah tím, že jim na něm bude více záležet, než kdyby dostali robotickou sadu a procvičovali si algoritmičké konstrukce a úlohy. Papert je zakladatelem jazyku LOGO, což je programovací jazyk, který vznikl koncem 60. let pro výuku ve škole. Seymour Papert a další ho využívali na MIT (Massachusetts Institute of Technology) k výuce matematického myšlení s dětmi. Obsahoval tzv. "turtle graphics", která spočívala v tvorbě jednoduché počítačové grafiky. Tato metoda se inspirovala želvou, která za sebou v písku táhla ocásek a vytvářela čáru, odtud název "turtle graphics".<sup>26</sup>



Obr. 2 – Konstrukce čtverce pomocí želví grafiky

Prvním programem v Logo nás obvykle učí kreslit čtverec. Začínáme jedním řádkem kódu: FORWARD 50. Tento příkaz posune „želvu“ o 50 pixelů dopředu. Teď se objevuje otázka: co dál? Jestliže nevíte, zkuste chodit ve tvaru čtverce. Nejprve se posunete dopředu o 5 kroků. Co musíte udělat potom? Otočíte se doprava o 90 stupňů. A co dál? Toto cvičení

25/ TOCHÁČEK, Daniel a LAPEŠ, Jakub. Edukační robotika. Praha: Pedagogická fakulta Univerzity Karlovy, 2012. ISBN 978-80-7290-577-5. s. 22.

26/ HEMMENDINGER, David. Logo. Online. Britannica. 2024. Dostupné z: <https://www.britannica.com/technology/Logo>. [cit. 2024-04-30].

27/ RIMMOND, Scott. What is it like to be a turtle? Online. Medium. 2017. Dostupné z: <https://medium.com/thinking-with-computers/what-is-it-like-to-be-a-turtle-829eabb129>. [cit. 2024-04-30].

můžete provádět, stejně jako to dělal Papert, s malými dětmi. Teoretickou možností, pokud nevíme, jak problém vyřešit, je forma CS Unplugged, kde si žák stoupne a jako „želva“ vyšlape tvar své dráhy do čtverce. Vyzkoušený algoritmus následně napíše.<sup>28</sup>

Tímto označujeme kognitivní konstruktivismus, druhým proudem je sociální konstruktivismus, jenž vychází z prací o sociální dimenzi učení dle psychologa Lva Semjonoviče Vygotského. Jeho zóna nejbližšího vývoje představuje rozdíl mezi aktuální úrovní výkonu (tedy současnou schopností dítěte řešit určitý úkol) a potenciální vývojovou úrovní. Tuto vzdálenost lze překonat za pomoci učitele, dospělého nebo vyspělejšího dítěte. Což dle mého přesně doplňuje konstruktivistický přístup, kde se učitel stává pomocníkem a umožní žákovi dosáhnout vědomostí mimo jeho aktuální úroveň. Piaget kladl větší důraz na procesy zrání, které podle něj umožňují učení. V kontrastu k tomu Vygotskij zastával opačný názor. Podle něj je to učení, které formuje psychický vývoj a určuje jeho směr.<sup>29</sup>

---

28/ Tamtéž

29/ Vztah vývoje a učení, školní vyučování, zóna nejbližšího vývoje. Online. Portál. 2004. Dostupné z: <https://nakladatelstvi.portal.cz/nakladatelstvi/aktuality/78984>. [cit. 2024-06-11].

## 2 Projekt pomocí konceptu STEM

### 2.1 STEM

Tento koncept či akronym, který zastřešuje technické disciplíny ve vzdělávacích oborech přírodní vědy (Science), techniky (Technology), inženýrství (Engineering) a matematiky (Mathematics), vznikl v 90. letech v USA. Metoda je zaměřena na vzdělávání studentů, kteří díky realizaci projektu propojující předměty, dokážou myslet nekonvenčním způsobem, objevují, přijímají rizika problému, zapojují se do experimentálního učení a tvůrčího řešení problému. Koncept podporuje vztah mezi jednotlivými disciplínami s důrazem na kritické a kreativní myšlení. Reprezentuje koncepci propojující vzdělávání a profesní oblast se snahou zvýšení produktivity. Jednotlivé předměty můžeme učit tradičním způsobem, podle specifík jednotlivých disciplín, nebo multidisciplinární formou. Metoda se zaměřuje na výsledky a orientuje se na řešení problémů z reálného světa. Pomocí tohoto konceptu můžeme odvodit filozofii polytechnického vzdělávání, což chápeme jako vzdělávání integrující přírodovědné, technické a enviromentální vzdělávání. Přispívá nejen k rozšíření poznatků, ale i k vytváření pracovních dovedností, které jsou využívány v pracovním životě. Rozvíjí spolupráci v týmu vzájemnou komunikací pro vyřešení problému. K dispozici mohou být dva přístupy ke spolupráci jednotlivých vzdělávacích oblastí. Buď můžeme spolupracovat individuálně v jednotlivých oblastech S. T. E. M., nebo vytvořit novou super-oblast STEM.<sup>30</sup>

Tento záměr pro výuku dokázal, že oblasti vzdělávání, tedy vzdělávací předměty, nemusí mít z principu didaktickou spojitost. Učivo, které spadá pod výuku provedenou touto metodou, může být rozvrstveno do dalších oblastí (předmětů), než do těch, které jsou pro naše školství typické. S nátlakem, že je školství zastaralé a nepřístupující k novým didaktickým metodám, je zapotřebí přijít se změnou pro uspokojení společnosti. Jednou variantou, jak tohoto cíle dosáhnout, je právě integrace. Koncept je vnímán jako integrační

---

30/ Koncepce dovedností STEM má přispět ke zvýšení mezinárodní konkurenceschopnosti Austrálie. Online. NPI. 2019. Dostupné z: <https://archiv-nuv.npi.cz/vystupy/koncepce-dovednosti-stem-ma-prispet-ke-zvyseni-mezinarodni.html>. [cit. 2024-03-10].

proces vázaný na předměty, které tento návrh naplňují. Pro české vzdělávací prostředí znamená metoda STEM kvalitativní změnu v pojetí vzdělávání a jeho vnitřním uspořádání.<sup>31</sup>

Podívejme se na jednotlivé oblasti a na aktivity, které mohou v těchto oblastech žáci plnit. STEM se může rozvést o další písmeno A (Art) a tedy vzdělávací koncept STEAM, který rozvíjí původní koncept o uměleckou neboli výtvarnou složku. V oblasti S (věda) žáci zkoumají a vypracovávají vědeckou část svého projektu, pomocí vědeckých experimentů, výpočtů a zkoumání mohou proniknout do problematiky projektu a poznávat pravidla fungování různých vztahů a jevů. Vědomosti získané z této fáze jsou potřebné pro konstrukci a celkové uchopení pravidel fungování projektu realizovaného žáky. Oblast T (technologie), zde žák využije různé typy nástrojů, jako jsou tablety, počítače a software k vytvoření technologických prototypů svého projektu. V této oblasti si žák rozvíjí digitální dovednosti. Typické pro oblast E (inženýrství) je tvoření konstrukčních prvků navrženého prototypu. V tvorbě žák pozná a využije různé druhy materiálů, jako je dřevo, plastelína, kov, papír, stavebnice a prvky pro 3D tisk. Oblast M (matematika), ve které žáci provádí matematické výpočty pro realizaci projektu. Tyto výpočty se často současně provádí již v konstrukční oblasti. Bez těchto výpočtů bychom nedošli ke správnému řešení a k optimalizaci softwaru nebo technologické stránky projektu. V neposlední řadě oblast A (umění), kde žáci vytvářejí uměleckou část svého projektu. V této oblasti mohou také navrhovat prototypy projektu, ať už jde o skici, nebo modely v různých softwarových prostředích.<sup>32</sup>

Rozvoj umění v konceptu STE(A)M nemusí být pouze umění výtvarné, hudební, nebo divadelní. Jedná se také o např. průmyslový, umělecký design. Oproti STEMu se nezaměřuje výhradně na vědecké koncepty v jednotlivých oborech. Využití se v dnešní době objevuje u tzv. hybridních zaměstnání, která kombinují vědu a umění. Mezi které spadá 3D modelář, animátor, produktový designer, či grafický designer. Toto rozšíření má samozřejmě i své

---

31/ Koncept STEM. Online. NPI. 2019. Dostupné z: <https://archiv-nuv.npi.cz/p-kap/koncept-stem.html>. [cit. 2024-03-10].

32/ STEM/MINT koncept vzdělávání 2023. Online. Inšgraf. 2023. Dostupné z: <https://insgraf.cz/katalog/katalog-STEAM-MINT-koncept-vzdelavani-2023/II/index.html>. [cit. 2024-03-10].

odpůrce. Podle Susan Singer (profesorka z Carleton College) sloučení vědy a umění snižuje hodnotu obou těchto oblastí.

Od konceptu se dostáváme k jeho aplikaci přímo ve škole. Koncept nezůstal nepovšimnutý u různých společností, které zajišťují nábytek, vybavení či jiné edukační potřeby do školy. Jednou z nich je firma Inigraf, která nabízí návrh a nábytek pro STEM učebny. Jejich pohled vychází z oblastí STEMu, takže učebnu rozdělí do jednotlivých oblastí a přichází s místem pro holistický rozvoj žáků s paralelním vzděláváním v rámci jednotlivých oblastí. Žáci jsou rozděleni do skupin tak, že pracují v každé zóně učebny, aby splnili veškeré úkoly ke zdárnému vypracování projektu. Tyto učebny s vyučovací metodou mohou být přínosné pro projektové vyučování, tedy např. pro náš projekt konstrukce vlastního robota.



33

*Obr. 3 – Návrh STEM učebny*

Počáteční vlnou byla opatření pro vysoké školství, následně úroveň středních škol, která připravuje absolventy na výkon povolání v těchto oblastech. Ukázalo se, že ke zvýšení

---

33/ What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/__data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].

kvality a počtu absolventů v oblasti STEM je potřeba začít s přípravou žáků základních škol. Ta sehrává podstatnou roli v profesní orientaci a pokládá základy znalostí, dovedností a postojů, které jsou klíčové pro nadcházející vzdělání. V prvním desetiletí 21. století a nyní můžeme sledovat vyšší pozornost v USA, Evropě či Austrálii. Zejména díky tomu, že ubývá počet studentů s narůstajícím nezájmem o studium těchto předmětů. Tyto obory jsou zrovna ty, co rozvíjí a podporují růst ekonomik. Proto sledujeme zásadní orientaci na vzdělávání, které je vnímáno jako zásadní faktor.

Podle světového hodnocení PISA v obrázku níže můžeme vidět státy, které si vedou v matematice lépe (zeleně) a hůře (červeně). Můžeme říct, že typ výuky se může promítnout na výsledcích PISA testování. Zrovna státy jako je Singapur, Čína, Kanada, Austrálie, Finsko, Spojené státy, Spojené království a Francie se nachází na horních příčkách žebříčku a využívají koncept STEM ve výuce. Obecně je výuka v těchto zemích více badatelská v porovnání s Českou republikou. V PISA testování z roku 2022 je průměrné matematické skóre mezi OECD zeměmi 472 bodů. Tyto země, tedy zbarvené zeleně, se umístily nadprůměrně. Česká republika se umístila na 21. místě se 486 body a tedy nadprůměrně.



Obr. 4 – PISA hodnocení 2022

34/ PISA 2022 Worldwide Ranking. Online. FactsMaps. 2023. Dostupné z: <https://factsmaps.com/pisa-2022-worldwide-ranking-average-score-of-mathematics-science-and-reading/>. [cit. 2024-03-20].



Pokud se budeme bavit o tom, které obory jsou přesně zahrnuty pod Science, tak neexistuje univerzální shoda. Tím je myšleno, zda věda zahrnuje sociální vědy, humanitní předměty, psychologii, ekonomii a politologii. Ve Spojeném království jsou kategorizovány do protějšku STEMu na humanitní a umělecké vědy, tak vznikla zkratka HASS (humanitní, umělecké a sociální vědy). Ta se v roce 2020 přejmenovala na SHAPE (společenské vědy, humanitní vědy, umění pro lidi a ekonomika). Národní centrum pro výzkum NCVER provedlo podrobný výzkum STEM dovedností a shrnulo výsledky v dokumentu What is STEM? Prokázali, že statistiky zaměstnanosti v oborech STEM mohou být problematické. Řekněme, že jsou velmi citlivé na úroveň vzdělání (např. certifikát, diplom bakalářský a vyšší). Data popisují, že míra zaměstnanosti se STEM kvalifikací je 4 % nebo 15 %, podle zvolené úrovně zaměstnání. Diplomovaných bakalářem či vyšším je samozřejmě menší procento, přesně 651 000. Vzhledem k těmto poznatkům není termín STEM vhodný pro statistické výkazy a debaty.<sup>35</sup>

### 2.1.1 The house of STEM

Abychom zlepšili porozumění a aplikaci konceptu, který jsme si představili, můžeme si představit analogii budovy s různými místnostmi na principu dekonstrukce. Ten je založen na modelu dovedností od Cunnighama a Vilasena (2010). Ten popisuje čtyři sady dovedností zaměstnavatele založených na literatuře z ekonomiky a psychologie a na globální metaanalýze zaměstnancových schopností.

Sady:

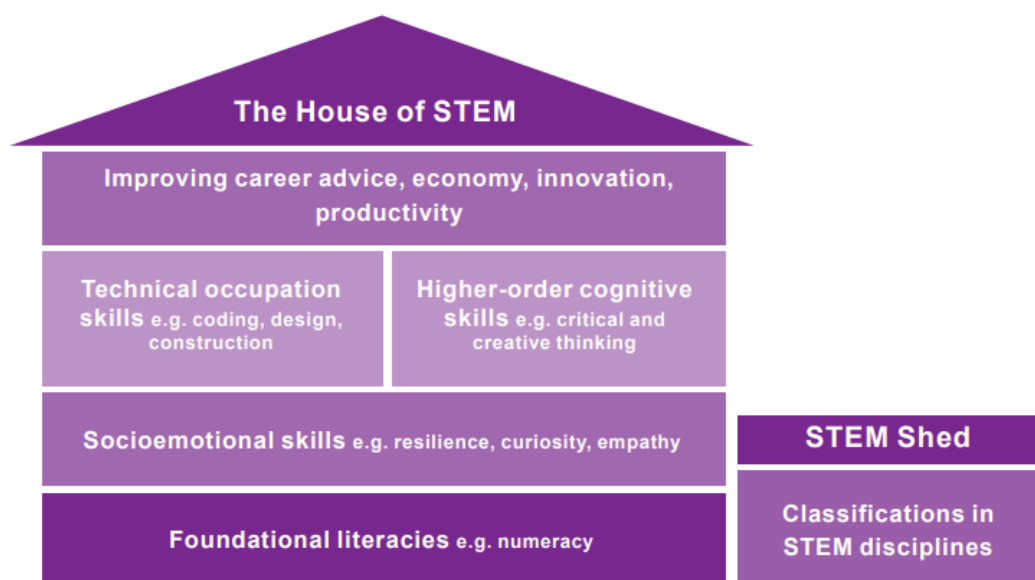
- socio-emocionální (např. odolnost)
- základní kognitivní (např. počítání)
- kognitivní vyššího řádu (např. kritické myšlení)
- technické dovednosti (např. kódování)<sup>36</sup>

---

35/ What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/__data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].

36/ What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016, s. 3–5. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/__data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].

Ve zmiňované analogii (viz obr. níže) je STEM střecha, která pokrývá budovu. Ta stojí na základech, mezi které patří schopnost počítat, číst a psát. Nad základy je místnost socioemocionálních schopností, mezi něž spadá zvědavost, odolnost a schopnost věřit ve své možnosti s motivací. Další místnosti jsou obsazeny pokročilými kognitivními schopnostmi, jako je kritické a kreativní myšlení, disciplína a technické dovednosti. Tyto předešlé místnosti podněcují k dalším inovacím, ke zvýšení produktivity a vedou k celkovému rozvoji žáka nebo ekonomiky státu, který vyhledává takové zaměstnance. Vedle naší budovy stojí STEM Shed (kůlna), představuje nástroje ke kategorizování disciplín, měření schopností, výstupů ve vzdělávání nebo v průmyslu.



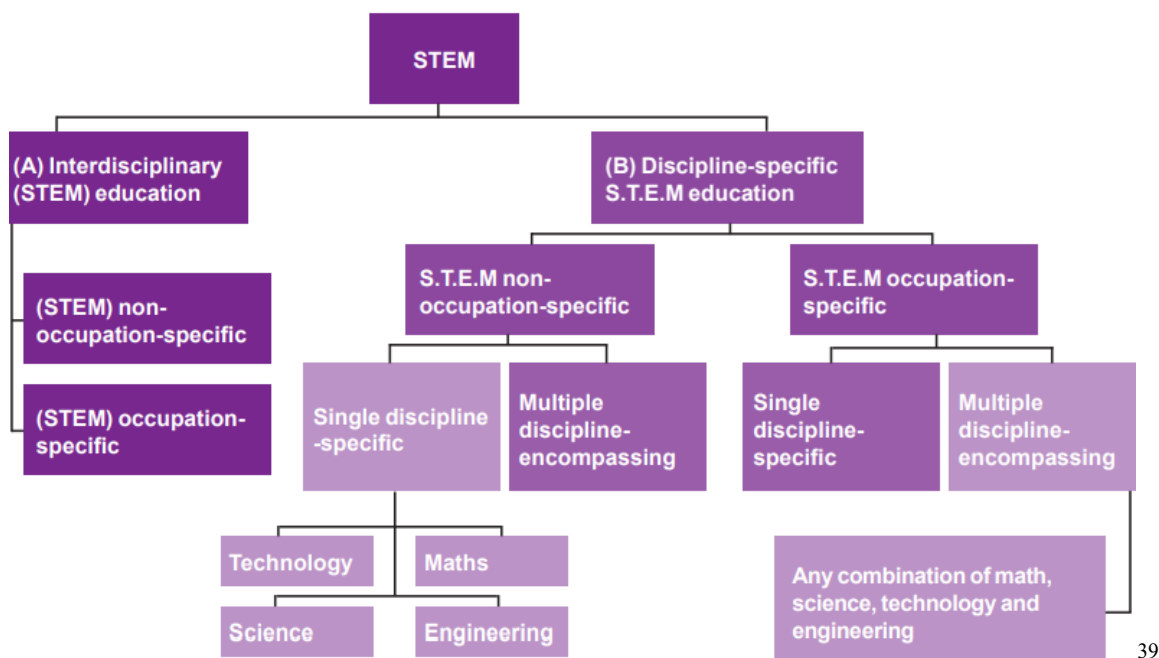
37

Obr. 5 – The House of STEM

Jak již bylo řečeno, STEM vzdělávání je inkluzivní ke společnosti jako k celku. Cílem tohoto vzdělávání je vědecká a technická gramotnost pro všechny, specifické profesní dovednosti pro vybrané povolání. Tyto výstupy mohou být docíleny buď pomocí interdisciplinárního a integrovaného vzdělávání, nebo „tradičně“ samostatně v jednotlivých disciplínách. Rozdíl mezi těmito přístupy se může zdát ne nijak zásadní, ale může mít však zásadní důsledky na učební kurikula. Interdisciplinární a integrované vzdělávání chápeme jako fúzi disciplín k pochopení a vyřešení problému ze života. K tomu musíme využít kritické a kreativní

myšlení, nebát se experimentovat a zkoumat. Přirovnat můžeme spolupráci doktorů z různých oborů (radiolog, kardiolog atd.), kteří spolu mohou dojít k novým závěrům.<sup>38</sup>

Tyto dva přístupy můžeme dále dělit podle toho, jestli se jedná o schopnosti specifické pro povolání (occupation-specific) a ty, které nejsou specifické pro povolání (non-occupation-specific). Tím dostaneme strom klasifikace vzdělávání pomocí STEM metody.



Obr. 6 – Schéma STEM

Závěrem lze konstatovat, že STEM metoda nabízí komplexní a interaktivní přístup k učení, který podporuje rozvoj klíčových dovedností. Aktivně zapojuje žáky konstruktivistickým přístupem do praktických projektů a experimentů, což podněcuje jejich zvědavost, kreativitu a analytické myšlení. Díky propojení vědeckých disciplín v rámci STEM přístupu mají žáci možnost chápat komplexní souvislosti mezi různými obory a lépe si uvědomovat praktické využití svých znalostí.

38/ What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016, s. 6–7. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/__data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].

39/ Tamtéž

Pro náš projekt je tento koncept zásadní a je na něm celý postaven. Nyní se podíváme na jednotlivé oblasti potřebné k realizaci projektu. Tedy pro návrh modelu 3D modelování, pro funkčnost robota si rozebereme mikrokontrolery Arduino a jejich možnosti programování.

## 2.2 3D modelování

Být úspěšným 3D grafikem znamená využívat všelijaké znalosti z různých oborů: kreslení, sochařství, architektura, inženýrství atd. 3D grafika vychází z vektorové 2D grafiky. Jediným rozdílem je, že se geometrická data neukládají pouze v rovině, ale v prostoru, třidimenzionální soustavě souřadnic. Práci ve 3D lze rozdělit na pět základních technologií (oblastí). Pokud animátor, grafik, respektive modelář pracuje sám, musí zvládat všechny technologie. Častěji však pracují ve společnostech, kde se tvorbou 3D grafiky zabývá více zaměstnanců. Typicky se každý z nich specializuje na určitou disciplínu.

<sup>40</sup> Základní technologie:

- Modelování – Tvorba objektů ve scéně.
- Texturování – Definování vlastností povrchu objektů.
- Osvětlení – Osvětlování scény a objektů.
- Animace – Tvorba animace pomocí klíčových snímků.
- Rendering – Proces převodu 3D modelů na 2D obrázky (rasterizace).

V neposlední řadě mezi další teoreticky nezbytné elementy mohou patřit vizuální efekty (exploze, kouř atd.), zvuk (hudba, zvukové efekty) a post produkce.

Naše práce se bude zabývat pouze konstrukcí robota modelováním pro vytvoření modelu robota v prostředí Tinkercad a následným vytisknutím na 3D tiskárně. Ostatním technologiím pro tvorbu 3D grafiky se prozatím věnovat nebudeme.

Abychom mohli modelovat, musíme se seznámit s navigací v prostoru. Standardně se využívá souřadnicový systém s mřížkou. Tento systém je tvořen třemi osami x, y, z. Jejich průsečík je středem, nebo také počátkem soustavy. Každá z těchto os má svoji kladnou a zápornou část, dělenou počátkem. Z reálného světa můžeme analogicky přiřadit jednotlivým

---

40/ Cinema 4D Release 6. Computer Press, 2004. ISBN 80-251-0001-4. s. 14.

osám následující vlastnosti. Osa  $x$  reprezentuje šířku, osa  $y$  výšku a osa  $z$  hloubku. Cokoliv od samotného objektu, jeho polohy či transformace, tedy animace, lze vyjádřit pomocí souřadnic  $X, Y, Z$ . Souřadnicové systémy dále můžeme dělit na globální a lokální. Globální představuje ten, se kterým jsme se seznámili. Lokální souřadnicový systém je přiřazen jednotlivým objektům, ty mají vlastní souřadnice a vlastní osy, kterými mohou být posunuty dovnitř objektu či kamkoliv do scény.<sup>41</sup>

### 2.2.1 Primitiva

Vytváření složitých modelů může být na první pohled náročné, proto lze využít tzv. blokového modelování. Objekty lze rozdělit na jednotlivé jednodušší části. Pokud se podíváme kolem sebe, zjistíme, že většina objektů se skládá z více základních tvarů. Tyto základní tvary nazýváme primitivy a jsou stavebními kameny mnoha modelů. Mezi základní primitiva spadají např. válce, kvádry, hranoly, koule, jehlany a kužely. Rovnice popisující primitiva jsou optimalizovány pro efektivní výpočet, což snižuje nároky na paměť RAM i na disk. Všechny primitivní objekty jsou parametrické, což znamená, že je možné je upravovat pomocí kontrolních bodů, což zjednodušuje jejich úpravu.<sup>42</sup>

Analogií k základní jednotce bitmapové grafiky, tedy pixelu, jsou tzv. polygony. Tyto plochy tvoří všechny objekty společně s vertexy (bod v prostoru) a hranami. Polygony jsou rozděleny podle počtu stran, z kterých se skládají. (3 – Triangle, 4 – Quadrilateral, 5 – Pentagon, 6 – Hexagon, 7 – Heptagon, 8 – Octagon, 9 – Nonagon a 10 – Decagon). Čím je větší počet polygonu, tím jemnější objekt dostaneme. Polygonový model má smysl pro práci a editaci modelu, jelikož vykreslení modelu je méně náročné než plně vystínovaný model, kde by delší renderovací časy zpomalovaly počítač a samotnou práci. Dohromady vytvoří tzv.<sup>43</sup> polygonovou síť, což je soubor vrcholů, hran a ploch, které definují tvar objektu ve 3D počítačové grafice a geometrickém modelování. Plochy jsou obvykle složeny z trojúhelníků, čtyřúhelníků nebo jiných jednoduchých konvexních polygonů, což usnadňuje

---

41/ Cinema 4D Release 6. Computer Press, 2004. ISBN 80-251-0001-4. s. 35.

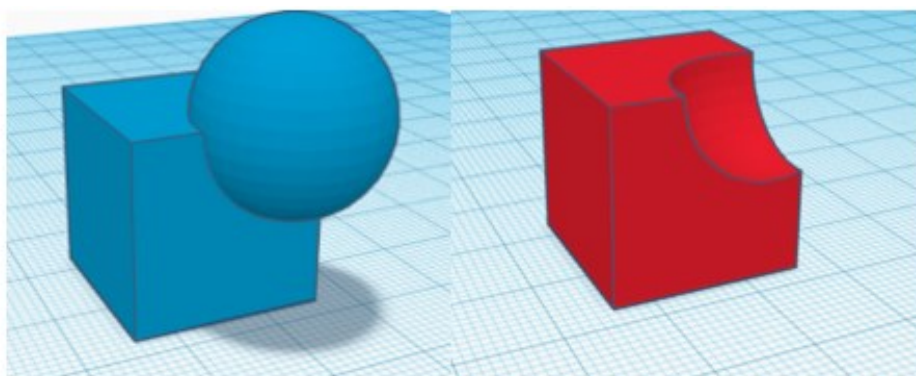
42/ Cinema 4D Release 6. Computer Press, 2004. ISBN 80-251-0001-4. s. 39.

43/ Polygonal Mesh. Online. ScienceDirect. 2018. Dostupné z: <https://www.sciencedirect.com/topics/computer-science/polygonal-mesh>. [cit. 2024-04-02].

vykreslování, ale mohou být také složeny z obecnějších konkávních polygonů nebo polygonů s otvory. Kromě polygonů lze v modelování používat také NURBS (Non-Uniform Rational B-Splines), tato metoda zde ale nebude probírána.

### **Booleovské operace**

Tyto operace, které jsou známé z matematiky, nacházejí své uplatnění i v modelování pro konstrukci složitějších objektů. Ve srovnání s matematikou, kde se tyto operace používají na množinách, v modelování se aplikují na dva objekty, které se vzájemně překrývají. Na základě typu operace jsou tyto objekty zpracovány a následně vytvoří nový objekt. Například v případě operace sloučení jsou vstupní objekty spojeny do jednoho objektu (sjednocení), zatímco operace odečtení vyřizne tvar jednoho objektu z druhého (rozdíl). Existují také další typy těchto operací, jako je průsečík (průnik dvou objektů) nebo operace podobná rozdílu, která zachová pouze vnější obal (plášť). Tinkercad může nabízet omezenou sadu těchto operací, ale v jiných profesionálních nástrojích je škála dostupných booleovských operací širší.<sup>44</sup>



*Obr. 7 – Booleovské operace ve 3D grafice*

### **2.2.2 Transformace**

V počítačové grafice využíváme k manipulaci objektů transformace založené na transformačních maticích. Skládání jednotlivých zobrazení analyticky využívá násobení matic zobrazení. Jelikož není násobení matic komutativní, tak platí, že záleží na pořadí provedených transformací. Transformační matice jsou aplikovány na bod reprezentovaný v homogenních souřadnicích. Celý objekt transformujeme užitím matice na všechny body.

---

44/ Cinema 4D Release 6. Computer Press, 2004. ISBN 80-251-0001-4. s. 41.

Pracujeme v prostoru, ale pro reprezentaci zvolíme matice v rovině, dvojrozměrně. Mějme bod  $P[X, Y]$  a bod  $P'[X', Y']$ , který vznikl transformací bodu vynásobením maticí. Představíme si matice zobrazení, které budeme s žáky využívat. Těmi bude translace (posunutí), stejnolehlost (škálování) a rotace. Ty pro naši práci v Tinkercadu vystačí, zároveň můžeme skládáním těchto transformací vytvářet složitější transformace.<sup>45</sup>

**Def.:** Geometrické zobrazení (Transformace)  $f: M \rightarrow M$  je transformací právě tehdy, když je bijektivní, tedy injektivní (prosté) a současně surjektivní (na).<sup>46</sup>

**Def.:** Necht'  $f, g$  jsou zobrazení  $E^2 \rightarrow E^2$ , která jsou popsána maticemi  $F, G$ . Pak zobrazení  $g \circ f$  je popsáno maticí  $G \cdot F$ .<sup>47</sup>

Transformační matice translace:<sup>48</sup>

Translace je afinním zobrazením, které zobrazuje každý bod na bod posunutý o samý vektor. Náleží mezi izometrická zobrazení (shodná). Nemá žádný pevný bod, proto využijeme homogenních souřadnic, kde vektor posunutí  $v = (v_x, v_y)$  je v homogenních souřadnicích  $v = (v_x, v_y, 1)$ . Obdobně pro body. Posunutí by analyticky mělo přičíst souřadnice vektoru k souřadnicím bodu neboli  $P' = P + v$ .

$$Tv = \begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix}. \text{ Pro třetí rozměr v prostoru } Tv = \begin{pmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Aplikace:

$$TvP = \begin{pmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x + v_x \\ P_y + v_y \\ P_z + v_z \\ 1 \end{bmatrix} = P + v$$

---

45/ ŽÁRA, Jiří; BENEŠ, Bedřich; SOCHOR, Jiří; FELKEL, Petr. Moderní počítačová grafika. Computer Press, 2005. ISBN 80-251-0454-0.

46/ KUŘINA, František. 10 geometrických transformací. PROMETHEUS, 2010. ISBN 80-7196-231-7.

47/ Tamtéž

48/ PAUL, Richard. Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. Cambridge, MA: MIT Press, 1981

Transformační matice stejnolehlosti: <sup>49</sup>

Škálování, změna velikosti objektu (scaling), patří mezi podobné zobrazení. Přesněji se jedná o stejnolehlost, ta je určena koeficientem stejnolehlosti  $k \in (\mathbb{R} \setminus \{0,1\})$ . Pokud nastane případ  $k=0$ , zobrazí se všechny body do počátku, což by znamenalo, že se nejedná o transformaci dle Kuřinovy definice, jelikož se nejedná o prosté zobrazení. Pokud  $k=1$ , tak se jedná o identitu, která není podobným zobrazením.

Je určena koeficienty podle souřadnicových os  $[Sx, Sy, Sz]$ . Středem je počátek soustavy  $[0, 0, 0]$ .  $P' = [X \cdot Sx, Y \cdot Sy, Z \cdot Sz, 1]^T$ .  $T$  v exponentu symbolizuje transponovanou matici, tedy představuje sloupec.

$$S = \begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Tím, že střed stejnolehlosti je v počátku, můžeme se zbavit 4. sloupce a tedy i 4. řádku, který pouze splní rovnost  $1=1$ . Dále zápis pro rotaci zjednodušíme tím, že se bude jednat o rotaci v počátku.

Transformační matice rotace:

Rotace je izometrickou transformací, která má vlastnost neměnicí vzdálenost mezi body. Pro rotaci kolem os podle počátku si vystačíme s maticemi zobrazení. Zajímavým pohledem v rovině je skutečnost rotace na jednotkové kružnici. Pokud si zobrazíme body na jednotkovou kružnici a provedeme rotaci, původní body se budou chovat stejně jako ty na kružnici. Pro tento způsob jsou velmi vhodná komplexní čísla, jelikož polární forma zápisu obsahuje úhel (měřený v počátku) již v definici. Vynásobením komplexního čísla  $z_1$  komplexním číslem  $z_2 = i$  získám rotaci o 90 stupňů. Komplexní čísla využijeme pouze v rovině, proto se zavedly kvaterniony  $q = a+bi+cj+dk$ , které mají tři imaginární složky ( $i, j, k$ ) a slouží pro rotaci v prostoru. Imaginární jednotky si můžeme představit jako jednotkové vektory ve směru souřadných os  $x, y$  a  $z$ , s tím, že  $b, c, d$  udávají souřadnice obecného 3D

---

49/ ŽÁRA, Jiří; BENEŠ, Bedřich; SOCHOR, Jiří; FELKEL, Petr. Moderní počítačová grafika. Computer Press, 2005. ISBN 80-251-0454-0.



vektoru vzhledem k těmto osám (skalár a reprezentuje reálnou složku kvaternionu). Rotace v prostoru funguje analogicky jako jednotková kružnice v rovině. Nadefinujeme ji na jednotkové sféře, všechny body v prostoru se budou chovat obdobně jako na sféře. Variantou je otáčet body podle os  $x$ ,  $y$ ,  $z$  podle počátku a následně zobrazení skládat s dalšími. Rotaci např. podle osy  $z$  značíme  $Rz$  a pro další  $Ry$ ,  $Rx$ .<sup>50</sup>

Rotace bodu kolem počátku soustavy je zadána úhlem  $\alpha$ . Nová pozice bodu po rotaci je určena  $P' = [X \cdot \cos(\alpha) - Y \cdot \sin(\alpha), X \cdot \sin(\alpha) + Y \cdot \cos(\alpha)]^T$ .

$${}^{51} Rz = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} Ry = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

$$Rx = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Právě možnost posunutí a skládání zobrazení nám dává sílu posunout si objekt do počátku a provést potřebné transformace. Následně objekt posunout zpět o vektor opačný.

Pro úpravy polohy os objektu se obvykle používají hodnoty  $X$ ,  $Y$ ,  $Z$ , což představuje posunutí v prostoru ve třech rozměrech. Na druhou stranu pro rotaci objektu se používají hodnoty  $H$ ,  $P$ ,  $B$ .  $H$  je zkratkou pro Heading (směr),  $P$  představuje Pitch (náklon nahoru/dolů) a  $B$  značí Bank (náklon vlevo/vpravo). V této reprezentaci je  $H$  obvykle reprezentováno hodnotou  $Ry$ ,  $P$  hodnotou  $Rx$  a  $B$  hodnotou  $Rz$ .<sup>52</sup>

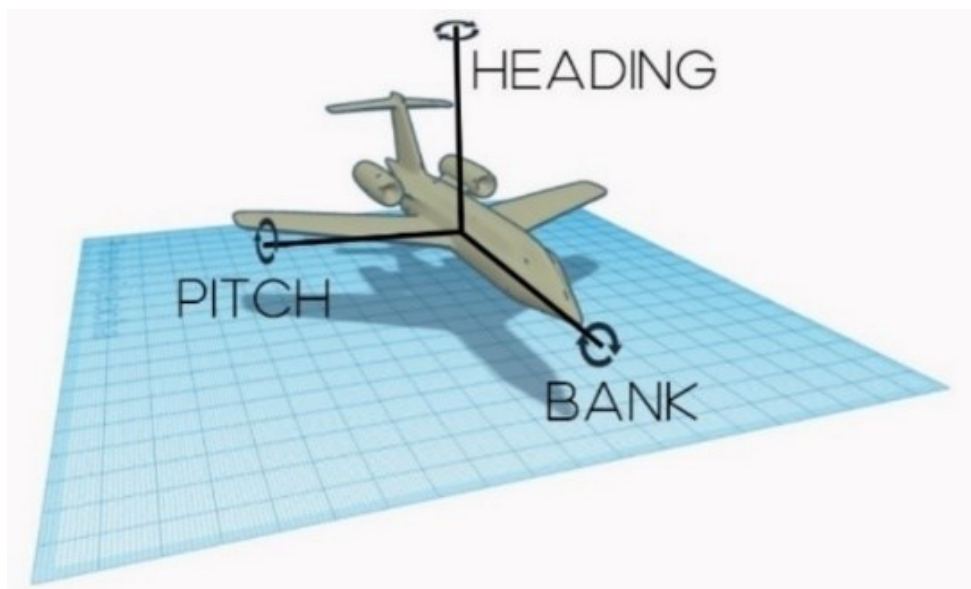
---

50/ ŘEHÁČEK, Jan. Jak zatočit s virtuálním světem. Online. Matfyz. 2019. Dostupné z: <https://www.matfyz.cz/clanky/matykani-xxvii-jak-zatocit-s-virtualnim-svetem>. [cit. 2024-04-02].

51/ Odvození transformačních matic pro různé typy rotace. Online. MUNI. 2013. Dostupné z: [https://mineralogie.sci.muni.cz/kap\\_1\\_3\\_symetrie/rotace\\_prikklad.htm](https://mineralogie.sci.muni.cz/kap_1_3_symetrie/rotace_prikklad.htm). [cit. 2024-04-02].

52/ Cinema 4D Release 6. Computer Press, 2004. ISBN 80-251-0001-4. s. 36.

Pro lepší představu termínů rotačních souřadnic si můžeme představit model letadla (viz obr. níže). Kladné hodnoty os  $x$  a  $y$  odpovídají rotaci proti směru hodinových ručiček, zatímco kladné hodnoty osy  $z$  odpovídají rotaci ve směru hodinových ručiček.



Obr. 8 – HPB souřadnice

## 2.3 Mikrokontrolery Arduino

Pro naše účely jsme se rozhodli použít open-source hardware s názvem Arduino. Tento open-source přístup zahrnuje software pro programování systému a hardware v podobě samotných mikrokontrolerů (Arduina). Jaký je rozdíl mezi mikrokontrolerem a mikroprocesorem? <sup>53</sup> Mikrokontroler je jednočipový počítač, který obsahuje v čipu CPU (central processing unit), RAM (random access memory), ROM (read-only memory) a vstupy a výstupy pro komunikaci s okolním světem. Naopak mikroprocesor má CPU integrované v jednom čipu, ale RAM, ROM a vstupy s výstupy jsou zpravidla samostatné součásti. Tento návrh může být využit ve vzdělávání k pochopení a ukázkám fungování různých systémů (například robotů, mikrovlákných trub a dalších nástrojů). Cílem je poskytnout programátorské koncepty desky Arduino UNO a představit její fungování a propojení s různými senzory a vstupně/výstupními zařízeními. Arduino není počítač ve formě stolního počítače nebo telefonu, ale umožňuje připojení různých senzorů, motorů,

---

53/ Co je to mikrokontrolér? Online. Zonepi. 2019. Dostupné z: <https://blog.zonepi.cz/maker-uno-ve-vyuuce-1-5-co-je-to-mikrokontroler/>. [cit. 2024-05-02].

diod atd. Programování probíhá na počítači v integrovaném vývojovém prostředí (IDE) a vytvořený kód je následně nahrán do Arduina, který je poté ovládán tímto programem. Získalo si popularitu díky své nízké ceně a dostupnosti softwaru. Pro práci s Arduinem jsou potřebné znalosti ve třech hlavních oblastech: První oblastí je porozumění hardwarové desce, druhou oblastí je seznámení s integrovaným vývojovým prostředím (IDE) a příkazy, které řídí hardware. Třetí oblastí je pak pochopení principu fungování senzorů a vstupně/výstupních zařízení, které slouží k získávání informací z okolí např. k reakcím robota.<sup>54</sup>

Elektřina je druh energie, podobně jako teplo, síla nebo světlo, a proudí vodiči. V elektrickém obvodu elektřina „teče“ z bodu s vyšší potenciální energií (obvykle označeného znaménkem +) do bodu s nižší potenciální energií. Zemnicí bod (označovaný znaménkem - nebo zkratkou GND) je obecně bodem s nejnižší potenciální energií v elektrických obvodech. V obvodech, které budete stavět, proudí elektrická energie pouze jedním směrem.<sup>55</sup>

### 2.3.1 Popis desky Arduino UNO

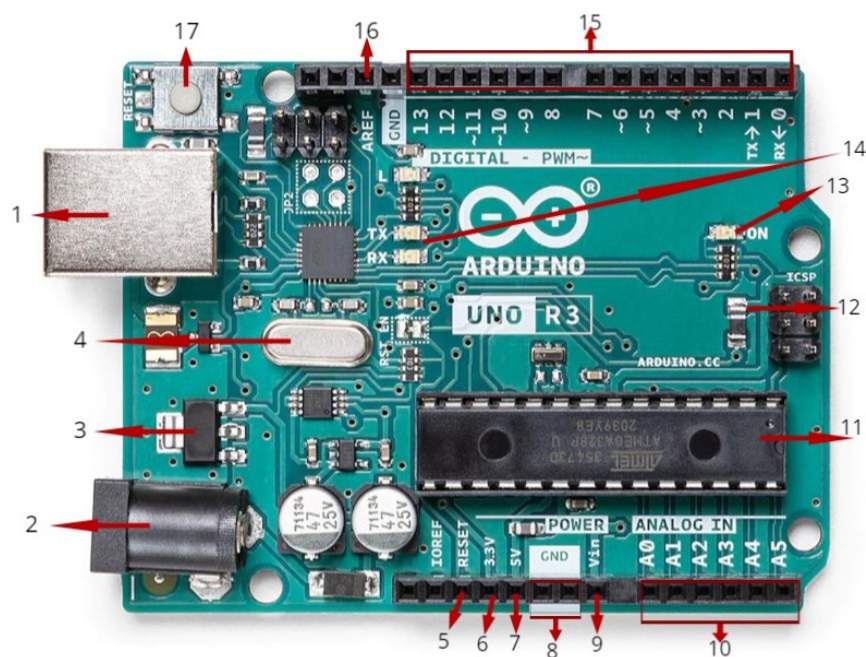
Pro naše účely zvolíme mikrokontroler Arduino UNO, který je založen na mikroprocesoru ATmega328P, který je hlavní částí „mozku“ desky. Tuto analogii můžeme rozšířit na celého robota, kde mikrokontroler představuje mozek robota, senzory jsou jeho oči a motory jsou zařízením pro pohyb robota. Tyto oblasti v této kapitole dále rozebereme. Ve školství je tato deska hojně používána svojí dostupnou cenou. Součástí sady je spousta součástek, my využijeme LED diody, krokové motory s řídicími deskami, bzučák, odpory, sonar na měření vzdálenosti, tlačítko a fotorezistor k měření intenzity světla. Jednotlivé součástky jako jsou žárovky, nebo reproduktory, jsou elektrickými měniči, ty mění různé formy energie na elektrickou a naopak. Jak už jsme zmiňovali analogii se senzory a motory, tak se k nim vrátíme. Součástky měnící el. energii na ostatní formy jsou pohony a obráceně z jiných forem na el. energii jsou senzory.

---

54/ MISRA, Yogesh. Programming and Interfacing with Arduino. CRC PRESS, 2021. ISBN 978-1032063164.

55/ NOVÁK, Milan a PECH, Jiří. Robotika pro střední školy: programujeme Arduino. Jihočeská univerzita v Českých Budějovicích, 2018. ISBN 978-80-7394-786-6.

Většina Arduino desek je stavěna podobně jako deska UNO na obr. níže.:



Obr. 9 – Popis Arduino desky

- (1) **Napájení USB** – varianta k napájení desky (5 V) a k nahrání kódu.
- (2) **Napájecí konektor** – náhrada místo USB, lze připojit ke střídavému proudu (220 V) pomocí adaptéru, vstupní napětí je 7–12 V.
- (3) Stabilizátor napětí – regulace stejnosměrného napětí.
- (4) Krystalový oscilátor – k synchronizaci operací mikroprocesoru a desky.
- (5) Reset – pin k restartu desky, pokud se program zasekne. Pokud je na pinu reset přivedeno napětí 0 V, deska Arduino se resetuje.
- (6) Výstup 3,3 V – pin pro napětí 3,3V.
- (7) **Výstup 5 V** – pin pro napájení elektrického obvodu s napětím 5 V (diskrétní).
- (8) **GND** – pin k napájení el. obvodu s napětím 0 V, k dispozici tři piny GND (ground).
- (9) Vin – desku lze zapnout připojením stejnosměrného napětí, v rozsahu 7-20 V na pin Vin, aby nedošlo ke zkratu.
- (10) **Analogové vstupy** – 6 pinů nabízející rozsah analogového vstupního napětí 0-5 V (spojitý).
- (11) Mikroprocesor ATmega328 – provádí logické a aritmetické operace.

- (12) ICSP – Piny chovající se jako AVR programátor, používány k programování nebo spouštění Arduina.
- (13) Power LED – při spuštění se dioda rozsvítí.
- (14) Tx a Rx LED – diody blikající při přenosu přes pin 1 Tx (transmit), nebo při příjmu na pinu 0 Rx (recieve).
- (15) **Digitální I/O** – 14 digitálních pinů, které mohou být buď vstupními, nebo výstupními piny.
- (16) AREF – využíván s analogovými piny, pokud je potřeba jiné napětí než 0-5 V.
- (17) **Reset tlačítko** – resetuje mikrokontroler. <sup>56</sup>

### 2.3.2 Dosavadní možnosti využití Arduino na 2. stupni ZŠ

Obecně můžeme říct, že STEM výuka využívající Arduino není v České republice příliš rozšířená. Pokud se Arduino využívá, tak často pouze v rámci jednoho předmětu, například ve fyzice k měření různých veličin pomocí senzorů nebo při výuce programování v informatice. Přesto existují školy, které tento způsob výuky uplatňují. Příkladem je článek od autorky Martiny Kupilíkové, která vyučuje robotiku pro žáky 9. ročníků. Využívala Arduino k výuce elektronických obvodů s LED diodami a tlačítky, které studenti nejprve zapojovali (viz obr. níže) a poté začali programovat. Paní učitelka zvolila Arduina Nano a žáci pracovali ve dvojicích na úkolech, jako je rozblikání jedné, nebo více diod. <sup>57</sup>

Druhým případem je ZŠ Tábořská, která také kombinuje výuku fyziky s Arduinem, tudíž můžeme říct, že splňuje multidisciplinární smysl koncepce STEM. Využívají ho k měření vzdálenosti, hmotnosti a teploty pomocí senzorů. Dále k zapojení základních elektrických obvodů jako je spínač nebo tlačítko a k propojení sériového a paralelního zapojení rezistorů. Provádí měření elektrického proudu, napětí a odporu rezistoru. S jeho pomocí lze ověřit Ohmův zákon, měřit výsledný odpor. <sup>58</sup>

---

56/ MISRA, Yogesh. Programming and Interfacing with Arduino. CRC PRESS, 2021. ISBN 978-1032063164.

57/ KUPILÍKOVÁ, Martina. Arduino na ZŠ. Online. Itfitness. 2020. Dostupné z: <http://itfitness.cz/arduino-na-zs/>. [cit. 2024-06-11].

58/ VOJTA, Václav. Arduino v 6. třídě. Online. Wix. 2024. Dostupné z: <https://vkp032.wixsite.com/arduino/fyzika>. [cit. 2024-06-11].

Jedním z komplexních příkladů STEM metody je taiwanská studie, ve které žáci byli vyučováni programování senzorově kontrolovaných projektů. Studie zahrnovala dvě skupiny, celkem 54 žáků. První skupina měla za úkol postavit "zelenou budovu", která díky senzorům reagovala na teplotu a dokázala automaticky otevřít střešní okna částečně nebo úplně. Druhá skupina sestavila zvíře, konkrétně žábu, která pomocí infračerveného senzoru rozpoznávala přítomnost jiných žab a při jejich spatření začala vydávat zvuky pomocí buzzeru, aby obhájila své území. Na těchto projektech se podíleli učitelé přírodních věd, informatiky, matematiky a dílen, jak je uvedeno v následující tabulce. <sup>59</sup>

TABLE I: DESIGN DESCRIPTION OF THE TWO-CLASS SUBJECT RESEARCH COURSE

teaching objectives	teaching content	teaching methods	teaching time	teaching teacher
1. Cross-disciplinary	Green building	Film teaching	2 lessons	information teacher
2. Students are interested	Animal beast	Question and answer		2 science teachers
3. sensory-controlled works	Test its function			
1. Arduino programming	RGB three-color LED light,	Demonstration teaching	2 lessons	information teacher
2. Solve problem of the sensor-controlled loop	buzzer, infrared sensor	Practice		2 science teachers
		Solve the problem		
1. Sensory work design	Design blueprint	Demonstration teaching	2 lessons	2 craft teachers
2. Select materials	Select base and material	Student report work content		2 science teachers
3. Select the base of works	Proportion and symmetry	Discussion and suggestion		
1. Assemble sense-controlled works	Each group assembly works	Assembly and testing	2 months, after	information teacher
2. test sense-controlled works	Each group test works	Discussion and suggestion	school time	2 craft teachers
		Rolling correction		2 science teachers

60

#### *Obr. 10 – Popis STEM projektu*

Pokud bychom odhlédli od ZŠ, tak imyšlení nabízí středoškolskou učebnici Arduino, kde např. úloha s papírovým robotem, kterého si žák nakreslí podle šablony, vystřihne a následně zabuduje do robota z papíru LED diodu a buzzer, vytváří prostředí k formě STEM výuky, kde žáci s programováním rozvíjí svou zručnost a kreativitu.

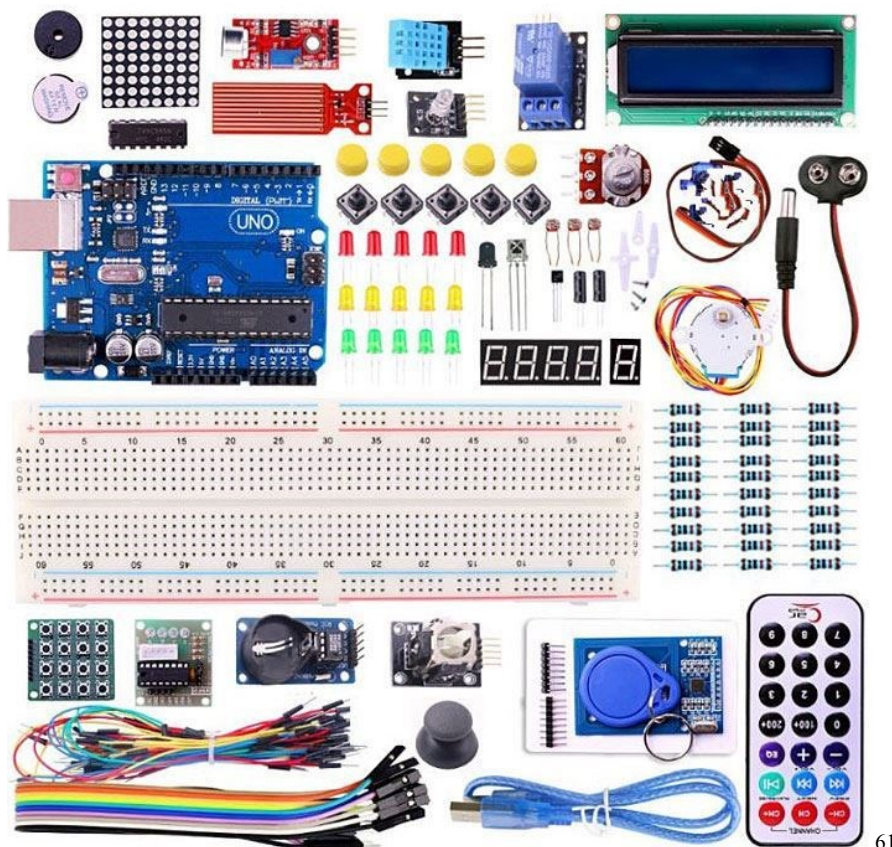
59/ LU, Chow-Chin. Elementary School Students Learn Arduino Programming to Assemble Sensory-Controlled Works. Online. 2020. Dostupné z: <https://www.ijiet.org/vol10/1374-IT006.pdf>. [cit. 2024-06-11].

60/ Tamtéž



### 2.3.3 Elektrotechnické součástky

Součástí sady Arduino jsou také různé elektronické komponenty určené pro tvorbu elektrických obvodů. Mezi něž patří nepájivé kontaktní pole, senzory, servomotory, krokové motory s řídicími deskami, LED diody, bzučáky, rezistory a propojovací kabely.



61

Obr. 11 – Elektronické součástky pro sadu Arduino UNO

Je důležité pochopit fungování nepájivých kontaktních polí (breadboards). Jedná se o jednoduchý nástroj pro vytváření el. obvodů, který umožňuje snadné propojování bez nutnosti pájení součástek. Na poli jsou červené a modré linky, které se vinou podél délky desky. Zdiřky nebo také piny vedle těchto barevně označených linek jsou určeny k použití jako napájecí 5 V (červené) a GND (modré) sběrnice. V těchto pinech vede drát vodorovně podél barevných čar. Celé pole je rozděleno do 4 oddělených bloků. Ve vnitřní části pole

61/ ARDUINO UNO R3 – Upgraded Learning Kit. Online. PS electronic. 2024. Dostupné z: <https://www.pselectronic.cz/index.php?action=k5505-155-arduino-uno-r3-upgraded-learning-kit>. [cit. 2024-05-02].

jsou naopak piny podloženy drátky svíse a slouží k zapojení jiných el. součástek a k tvorbě el. obvodů.<sup>62</sup>

Další velmi užívanou součástí jsou LED diody, polovodičové komponenty, které generují světlo, a proto se často používají k signalizaci nebo osvětlení. Charakteristickým rysem diod je schopnost propouštět elektrický proud pouze v jednom směru. Každá dioda se skládá z anody (+) a katody (-), přičemž delší nožička diody reprezentuje anodu. Na rozdíl od jiných světelných zdrojů jako jsou žárovky nebo výbojky, LED diody pracují s nižšími hodnotami proudu a napětí.<sup>63</sup> Odpor neboli rezistor je pasivní elektrotechnickou součástí, kterou využijeme ke snížení proudu v obvodu přímo ze vzorce  $U = R \cdot I$ , aby nedošlo k přetížení a zkratu v součástce. Dále pasivní bzučák, umožňuje vytvářet zvukové signály pomocí signálu napětí konkrétní frekvence. Tyto součástky propojujeme pomocí kabelů, které jsou ve tvaru samec/samice.

Robot se skládá z několika subsystémů, z nichž každý plní určitou funkci. Mezi ně patří akční (pohony), řídicí (mikrokontrolery), senzorické, komunikační a napájecí subsystémy. Komunikační a napájecí jsme zmínili u samotného mikrokontroleru, proto se podíváme na pohony a senzory.

### **2.3.4 Pohon – krokový motor**

V automatizaci se využívají 3 typy motorů elektrické, pneumatické a hydraulické. My využijeme krokový motor, ten je typem stejnosměrného elektromotoru, který transformuje plnou rotaci do několika individuálních kroků. Každý krok má pevně stanovenou velikost, což umožňuje přesné otočení motoru do specifické polohy s přesným úhlem. Krokový motor se skládá ze zmagnetizovaného rotoru a statoru s předem určeným počtem elektromagnetických cívek. Když jsou cívky napájeny elektrickým proudem, vytvářejí magnetická pole, která buď přitahují, nebo odtahují segmentovaný zmagnetizovaný rotor, což způsobuje jeho otáčení. Stator je tvořen plechy s drážkami, v nichž je umístěno statorové

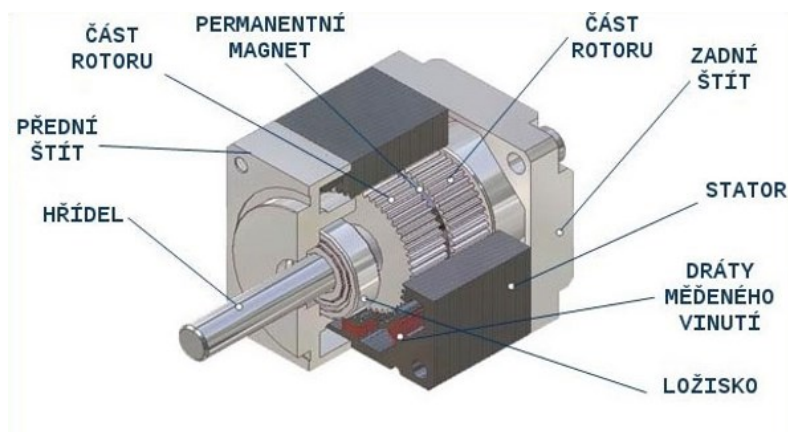
---

62/ BLUM, Jeremy. Exploring Arduino: Tools and Techniques for Engineering Wizardry. 2. John Wiley, 2020. ISBN 978-1-119-40535-1.

63/ CO JE TO LED DIODA? Online. LEDme. 2024. Dostupné z: <https://ledme.cz/textove-novinky/clanky/co-je-led-dioda>. [cit. 2024-05-02].



vinutí. Na druhé straně rotor může být tvořen buď železným jádrem (kde se po obvodu střídá zub/drážka), nebo permanentními magnety (na obvodu se z magnetů střídá sever a jih). Krokové motory jsou vhodné pro aplikace, které vyžadují přesné nastavení polohy a udržení této polohy i přes působící síly, jako jsou souřadnicové zapisovače nebo počítačem řízené obráběcí stroje. Jednoduchost a preciznost pohybu je činí v robotice velmi populárními. Nicméně je důležité zmínit i jejich nevýhody. Velmi zásadní je trvalý odběr proudu i v době, kdy se motor netočí, což může způsobit zbytečné energetické ztráty.<sup>64</sup>



65

Obr. 12 – Krokový motor

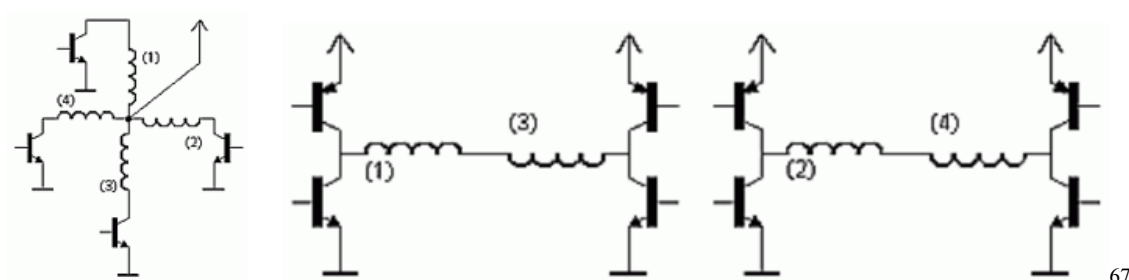
### Princip a metody řízení krokového motoru

Princip fungování krokového motoru je založen na proudu procházejícím cívkou statoru, kdy vzniká magnetické pole, které přitahuje opačný pól magnetu rotoru. Pokud vhodně zapojíme cívky, lze dosáhnout rotujícího magnetického pole, které motor pohání. Laicky řečeno, pokud prochází cívkou proud, vzniká magnetické pole, které přitahuje magnet, a tak dochází k pohybu. Existuje více variant metod řízení krokového motoru, a to podle požadovaného krouticího momentu. Typy řízení se dělí na unipolární, které využijeme s naším krokovým motorem pro naši konstrukci, a řízení bipolární. Při unipolárním řízení prochází proud právě jednou cívkou. Sice má nejmenší odběr, ale zároveň nejmenší krouticí

64/ Jak funguje krokový motor? Online. Raveo. 2024. Dostupné z: <https://www.raveo.cz/aktualita/jak-funguje-krokovy-motor/>. [cit. 2024-04-28].

65/ KROKOVÝ MOTOR – CO TO JE A JAK FUNGUJE. Online. EUfactory. 2018. Dostupné z: [https://shop.eufactory.com/blog/12\\_krokovy-motor-uvod](https://shop.eufactory.com/blog/12_krokovy-motor-uvod). [cit. 2024-04-28].

moment. Druhá varianta docílí vyššího krouticího momentu za cenu vyšší spotřeby. Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami, které jsou zapojené tak, aby měly navzájem opačně orientované magnetické pole, což docílí zrychlení pohybu. Dále řízení dělíme na jednofázové a dvoufázové. Při jednofázovém řízení generuje magnetické pole pouze jedna cívka motoru, nebo v případě bipolárního buzení dvojice cívek. Jak již z názvu vyplývá, dvoufázové řízení generují shodně orientovaná magnetická pole vždy vedle sebe sousedních cívek. To opět samozřejmě navýší na krouticím momentu, ale zdvojnásobí spotřebu oproti druhé variantě. Tyto metody dohromady tvoří tzv. řízení s polovičním krokem, což je technicky střídání kroků jednofázové a dvoufázové, čímž dosáhneme dvojnásobné přesnosti. Oproti tomu řízení plným krokem znamená, že na jednu otáčku je potřeba přesně tolik kroků, jako je zubů statoru.<sup>66</sup>



Obr. 13 – Dělení krokových motorů

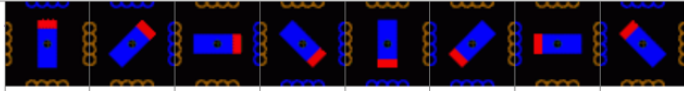
Pro našeho robota využijeme motor s unipolárním řízením s polovičním krokem. Na následujícím obr. máme motor se 4 cívkami a tabulku, která určuje, jakými cívkami prochází proud, a jakými ne. Cívky nabarvené hnědě, v tabulce nastavené hodnotou „0“, jsou bez proudu. Modře zbarvené cívky představuje magnetické pole, přitahuje červený konec magnetu (rotoru).

66/ ŘEZÁČ, Kamil. Krokové motory. Online. Robotika. 2002. Dostupné z: <https://robotika.cz/articles/steppers/cs>. [cit. 2024-03-27].

67/ Tamtéž

Pro jedno otočení tedy využijeme osm těchto kroků, které ale nebudou jednou reálnou otočkou na výstupu motoru, jelikož uvnitř motoru jsou ozubená kola fungující jako převod. Pro jednu otočku s našim motorem je potřeba těchto osm kroků provést 512krát. K výpočtu musíme znát počet fází statoru (ozn.  $m$ ) a počet zubů rotoru (ozn.  $n$ ). V našem případě má motor  $m = 64$  zubů rotoru a  $n = 45$  fází statoru. Vztah pro velikost kroku:

$$\Phi = \frac{360^\circ \cdot n}{m}$$



Cívka 1	-	-	0	0	0	0	0	-
Cívka 2	0	-	-	-	0	0	0	0
Cívka 3	0	0	0	-	-	-	0	0
Cívka 4	0	0	0	0	0	-	-	-

Obr. 14 – Proces jedné otočky krokového motoru

### 2.3.5 Senzory

Senzory v robotice představují klíčovou součást, která umožňuje robotům vnímat své okolí a provádět s ním interakce. Jejich použití přináší několik výhod, včetně zvýšení univerzálnosti a flexibility robotů, poskytování zpětné vazby pro adaptaci na prostředí a zajištění robustního chování v různých podmínkách. Senzory se v robotice využívají pro měření různých parametrů, jako je poloha, orientace, nebo detekce překážek a chybových stavů. Jejich třídění může probíhat podle mnoha hledisek, včetně měřené veličiny, fyzikálního principu, použité technologie, způsobu přenosu energie a typu využívané energie. Díky nim jsou roboti schopni úspěšně navigovat a interagovat s prostředím, ve kterém operují.<sup>69</sup>

V naší sadě využijeme senzor na měření vzdálenosti, konkrétně typ sonaru, který je založen na měření doby letu ultrazvukového impulsu. Senzor chvíli vysílá vlnu a pak přijímá vlnu odraženou od překážky, mikrokontroler spočítá časový rozdíl a vyhodnotí vzdálenost.

68/ Tamtéž

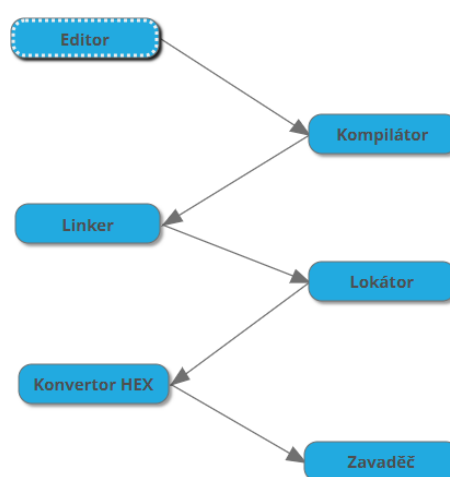
69/ HLAVÁČ, Václav. SENZORY PRO ROBOTIKU. Online. ČVUT. 2024. Dostupné z: <https://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/51SenzoryRobotika.pdf>. [cit. 2024-05-02].

Další senzory v naší sadě mohou být určeny pro měření různých veličin, jako je teplota, tlak nebo světelnost. Tyto senzory mohou být založeny na různých fyzikálních principech, jako jsou odporové, piezoelektrické, chemické, indukční nebo kapacitní. Dále je možné senzory rozdělit na aktivní a pasivní. Aktivní senzory fungují jako zdroje elektrické energie a reagují na změny snímané veličiny, zatímco u pasivních senzorů je nutné elektrickou veličinu dále zpracovat a vyžadují externí napájení.<sup>70</sup>

## 2.4 Programování mikrokontroleru Arduino

K vývoji programů se využívá stejnojmenné IDE Arduino, které po instalaci propojíme s Arduino deskou. Po nainstalování ovladačů, zvolení námi vybrané desky a správně vybraného portu (u Windows COM) můžeme programovat. IDE je založeno na vyšším programovacím jazyce Embedded C (rozšíření jazyku C pro tvorbu vestavěných systémů, programování mikrokontrolerů).

Skládá se z jednotlivých komponent: Editor, Kompilátor, Linker, Lokátor, Konvertor HEX a Loader (zavaděč). V Editoru píšeme program pomocí příkazů ve vyšším jazyce tvořící soubory s příponou .ino pro Arduino a .C/C++ pro jazyk C. Kompilátor následně převede program z vyšší úrovně do binárního jazyka. Kompilátor vygeneruje objektový soubor (.o) v Arduino a obdobně pro C/C++ soubor .obj. Dále je úkolem Linkeru spojit více než jeden objektový soubor stejného projektu do jednoho objektového souboru. Poté Lokátor přiřadí instrukcím adresu paměti kódu a vygeneruje absolutní objektový soubor. Ten převezme Konvertor HEX a převádí jej do hexadecimálního kódu s příponou .hex. Nakonec Loader zavádí hexadecimální kód do cílového řadiče a spouští program.<sup>71</sup>



Obr. 15 – Rozklad Arduino jazyku

70/ ŠŤASTNÝ, František. Senzory. Online. Amper MUNI. 1997. Dostupné z: [https://amper.ped.muni.cz/jenik/nejistoty/html\\_tree/node16.html](https://amper.ped.muni.cz/jenik/nejistoty/html_tree/node16.html). [cit. 2024-05-02].

71/ MISRA, Yogesh. Programming and Interfacing with Arduino. CRC PRESS, 2021. ISBN 978-1032063164.

### 2.4.1 Struktura programu

Když budeme mluvit o souborech, tak Arduino je pojmenovává jako „Sketch“. Základní strukturou každého programu jsou dvě funkce `setup()` a `loop()`. Je nutné v programu obě uvést, bez nich by se program nepřeložil a nespustil. Standardně se před těmito funkcemi na začátku programu deklarují proměnné.

```
void setup() {  
    // tělo funkce, může obsahovat příkazy (dvě lomítka tvoří komentář v rámci řádku)  
}  
  
void loop() {  
  
}
```

Funkce `setup()` je volána při spuštění programu, spustí se pouze jednou, a to po každém zapnutí nebo resetování desky. Slouží k inicializaci proměnných, nastavení režimů jednotlivých pinů (I/O), načtení knihoven, nastavení sériové komunikace a další inicializaci.

Funkce `loop()`, jak může název napovědět, provádí kontinuálně kód obsahující ve svém těle, dokud je deska Arduino napájena.

Většina pinů Arduina může sloužit jako digitální vstup nebo výstup. Směr pinu lze kdykoli změnit pomocí funkce `pinMode`, což znamená, že jeden pin může být v jednu chvíli vstupem a později výstupem. Digitální piny interpretují napětí jako dvě logické hodnoty: HIGH a LOW, také známé jako logická jednička a nula. U desky Arduino UNO logická hodnota HIGH odpovídá napětí blízkému napájecímu napětí, tedy 5 V, zatímco logická hodnota LOW odpovídá napětí blízkému 0 V.<sup>72</sup>

Syntaxe jazyka pro Arduino je velmi podobná syntaxi programovacího jazyka C++. Základní stavební jednotkou jsou příkazy, které mohou být přiřazovací, který něco počítá a tedy přiřazuje (`obsah_ctverce=a*a`), volání funkcí (`krok_dopredu()`) nebo jiné algoritmické konstrukce (podmínky, cykly). Každý příkaz je ukončen středníkem, ať už jde o deklarace

---

<sup>72/</sup> NOVÁK, Milan a PECH, Jiří. Robotika pro střední školy: programujeme Arduino. Jihočeská univerzita v Českých Budějovicích, 2018. ISBN 978-80-7394-786-6.

proměnných nebo jednotlivé příkazy ve funkci. Vše je v Arduinu case sensitive od funkcí, proměnných, po názvy knihoven, záleží tedy na velikosti písmen. Příkazy lze seskupovat do bloků, které se pak chovají, jako bychom na tom samém místě místo jednoho příkazu zapsali celý blok. Tato funkcionalita je užitečná zejména při vyhodnocování podmínek, pokud je podmínka splněna, lze vykonat celý blok příkazů místo pouhého jednoho. Bloky příkazů jsou odděleny složenými závorkami {}, dále také tvoří tělo funkcí.<sup>73</sup>

Velmi podstatnou věcí je přehlednost a srozumitelnost kódu buď ze svého hlediska, abych nezapomněl, co jaká proměnná znamená, nebo i z pohledu didaktického, kde proměnná „p“ a „pivot“ lépe vysvětlí, co v programu představuje. Tato srozumitelnost je klíčová zejména při spolupráci v týmu, kde různí lidé mohou pracovat na jednom kódu a je důležité, aby byl pro všechny jasný a snadno čitelný.

Všechny proměnné musí být deklarovány před jejich prvním použitím. Deklarace proměnné zahrnuje definování jejího typu (např. int, char, float), přiřazení jména a volitelně i počáteční hodnoty. Kromě proměnných máme také konstanty, které slouží k pojmenování hodnot, jež se během provádění programu nemění.<sup>74</sup>

#### 2.4.2 Funkce

Funkce je blok příkazů obsahující ve svém těle funkce, které se po jejím zavolání provedou. Před voláním funkce musí dojít k deklaraci funkce, ty mohou být s návratovou hodnotou, nebo bez návratové hodnoty typu void. Funkce může obsahovat parametry, které při volání konkrétně volíme. Logicky volíme datový typ podle toho, co funkce vrací. Typicky ve tvaru datový typ, název funkce (parametry) {tělo funkce}. Funkce void je bez návratové hodnoty, ale parametry obsahovat může.

Příkazy:<sup>75</sup>

---

73/ ARDUINO – příručka programátora. Online. DPS. 2021. Dostupné z: <https://www.dps-az.cz/getFile/id:84163/Arduino-p%C5%99%C3%ADru%C4%8Dka-program%C3%A1tora.pdf>. [cit. 2024-05-03].

74/ Tamtéž

75/ MISRA, Yogesh. Programming and Interfacing with Arduino. CRC PRESS, 2021. ISBN 978-1032063164.

**pinMode**(pin, MODE) – slouží ve funkci setup() k nastavení určeného pinu jako vstupního, nebo výstupního. Má dva parametry: pin, který určuje číslo pinu, a režim, který může být nastaven jako INPUT (vstup) nebo OUTPUT (výstup).

**digitalWrite**(pin, VALUE) – používá se k nastavení stavu určeného pinu, který byl předtím nastaven jako výstup pomocí funkce pinMode. Má dva parametry: pin, který určuje číslo pinu, a hodnotu, která může být buď HIGH (logická 1), nebo LOW (logická 0). Buď pinem proud protéká, nebo ne.

**digitalRead**(pin) – čte digitální hodnotu z určeného pinu, který je specifikován jako argument funkce. Tato funkce vrací hodnotu HIGH (5 V/logická 1), nebo LOW (0 V/logická 0).

**analogRead**(pin) – čte analogovou hodnotu z určeného analogového pinu, který je specifikován jako argument. Arduino deska obsahuje šest analogových pinů označených A0, A1, A2, A3, A4 a A5, které jsou připojeny k 10bitovému analogově-digitálnímu převodníku. Funkce analogRead vrací hodnotu v rozsahu 0-1023 odpovídající analogové hodnotě na pinu.

**analogWrite**(pin) – nastaví hodnotu pulzně šířkové modulace (PWM) na výstupní pin, což vytváří efekt pseudo-analogového signálu. Na desce Arduino Uno a podobných modelech lze k tomuto účelu využít piny 3, 5, 6, 9, 10 a 11.

**delay**(value) – pozastaví program na určitou dobu podle parametru v milisekundách (1s = 1000 ms).

if (podmínka) {příkaz, či blok příkazů} else {příkaz, či blok příkazů} – podmínky umožňují reagovat na aktuální situace tím, že rozhodují, která část kódu se má vykonat na základě toho, zda je podmínka splněna, nebo nesplněna.

for (int i = 0; i <n; i++) {} – cyklus pro předem určený počet opakování, který je ve tvaru (proměnná, podmínka a příkaz) {tělo cyklu}. Pokud je splněná podmínka, provede se tělo cyklu. Příkaz i++ přičte k i číslo 1.

### 3 Empirická část

Tím, že je robotika interdisciplinárním oborem, můžeme ji využít k výuce několika předmětů: informatiky, matematiky, fyziky a dílen. Z pohledu informatiky se zaměřujeme na rozvoj dovedností ve vzdělávacích oblastech algoritmizace a programování a digitálních technologií, zahrnujících práci s 3D tiskárnami, Arduinem jako hardwarem i softwarem a 3D modelovacím programem Tinkercad. Cílem je ověřit, zda jsou žáci schopni sestavit robotické auto. Tento proces zahrnuje modelování, 3D tisk, sestavení, zapojení el. součástek, až po naprogramování funkcí pro pohyb a primitivní inteligenci založenou na senzorech.

V tomto procesu kompletní konstrukce robota se zapojují předměty jako fyzika a matematika, které můžeme integrovat pomocí STEM výuky. Místo toho, abychom jednotlivé oblasti vyučovali odděleně, můžeme je spojit do jednoho celku. Výzkumy ukazují, že si lidé lépe pamatují věci v souvislostech. Proč tedy např. vyučovat Ohmův zákon samostatně, když ho můžeme využívat pro výpočet potřebného odporu v obvodu? Spatřuji v tomto STEM přístupu větší smysl i z hlediska motivace. Žáci jsou více motivováni vyřešit úlohu s odporem, aby mohli pokročit v celém robotickém projektu, než kdyby pouze počítali úlohy podle vzorce a učili se jej nazpaměť. Což souvisí i s konstruktivistickým pojetím výuky, které je pro tyto předměty podstatné.

Cílem empirické části je popis realizace stavby a naprogramování robota, navržení námětů úloh pro STEM výuku a provedení a analyzování případové studie na ZŠ Na Lukách v Poličce s žáky 9. třídy, kteří touto výukou projdou. Kvalitativní výzkum zahrnuje pozorování během hodin, rozhovory s žáky, dotazníky, pretest a posttest. Test se skládá z úloh zaměřených na matematiku, fyziku a informatiku. Zjednodušeně otázkou je, zda je schopen žák projít projektem, který je určen spíše pro střední školy.



### 3.1 Výzkumná otázka

**VO<sub>1</sub>:** Jaká je úspěšnost a efektivita žáků při řešení navržené sady konstruktivisticky orientovaných robotických úloh zahrnujících návrh, sestavení a programování robotů využívajících systém Arduino?

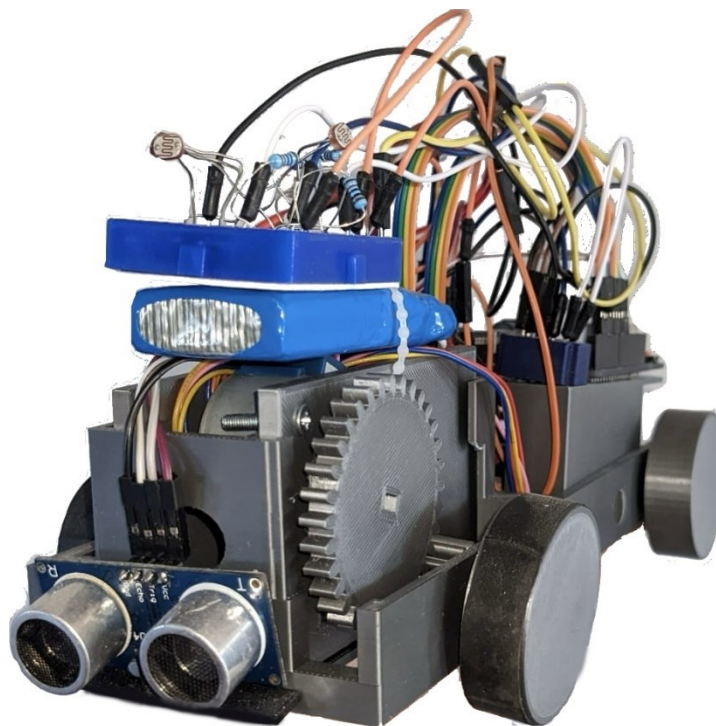
**VO<sub>2</sub>:** Dokážou žáci identifikovat a srozumitelně formulovat konkrétní problémy a obtíže spojené s řešením úkolů, jež jsou součástí řešitelem připravené sady úloh?

**H<sub>1</sub>:** Žáci 9. ročníku ZŠ budou schopni sestavit a naprogramovat robotický systém s využitím Arduina a disponují dostatečnými znalostmi a dovednostmi řešit typ úkolů, které jsou součástí řešitelem připravené sady úloh.

**H<sub>2</sub>:** Žáci budou schopni identifikovat konkrétní problémy a obtíže, které vyvstanou při řešení interdisciplinárních úloh v rámci řešitelem navrženého robotického projektu.

**H<sub>3</sub>:** Realizace řešitelem navrženého robotického projektu s využitím Arduina povede k dosažení stanovených očekávaných výstupů ve STEM předmětech.

### 3.2 Robotické auto



*Obr. 16 – Robotické Arduino Auto*

Tento projekt má za cíl prezentovat demonstraci výuky informatiky, rozvíjení informatického myšlení, programování, algoritmizace a dekompozice problémů v 9. ročníku prostřednictvím robotiky v jednotlivých oblastech STEM výuky. Co všechno by toto robotické zařízení mělo zvládat? Mělo by se pohybovat pomocí pohonu s krokovými motory, využívat senzor vzdálenosti k vyhýbání se překážkám, a nakonec by mělo pomocí fotorezistorů následovat zdroj světla. Robotika jako interdisciplinární oblast zahrnující elektroniku, programování a mechaniku umožňuje využití STEM přístupu k výuce. Žáci se zlepšují ve všech těchto oblastech současně, a zároveň mohou být motivováni samotným procesem tvorby robota. Práce s fyzickým třírozměrným objektem byla z pohledu vzdělávání shledána efektivnější než práce s abstraktními programátorskými úlohami, což usnadňuje a urychluje proces učení. Roboti mohou podporovat kontextové myšlení, protože nutí žáky přemýšlet v souvislostech učebního procesu. Žáci obdrží návrh podvozku robota, který prošel několika iteracemi návrhu, sestavení, analýzy a redesignu. Tento proces je časově náročný, protože každá verze odhalí nové problémy a samotný tisk je velmi zdoluhavý, což by vedlo k neefektivnímu využití času při návrhu.

Jednotlivé oblasti můžeme propojit s konkrétními kroky v procesu konstrukce robota. Navržení 3D modelu robota spadá do oblasti vědy (Science) a technologií (Technology), tisk na 3D tiskárně, sestavení, zapojení elektronických součástek a pájení do oblasti inženýrství (Engineering). Výpočty patří do matematiky (Mathematics) a fyzikální vztahy a výpočty do vědy (Science). Programování pak náleží do oblasti technologií (Technology). Tento projekt následně vytváří sadu dílčích úkolů pro žáky, které je provedou vlastním sestavením a zprovozněním robota.

### **3.2.1 Navržení 3D modelu**

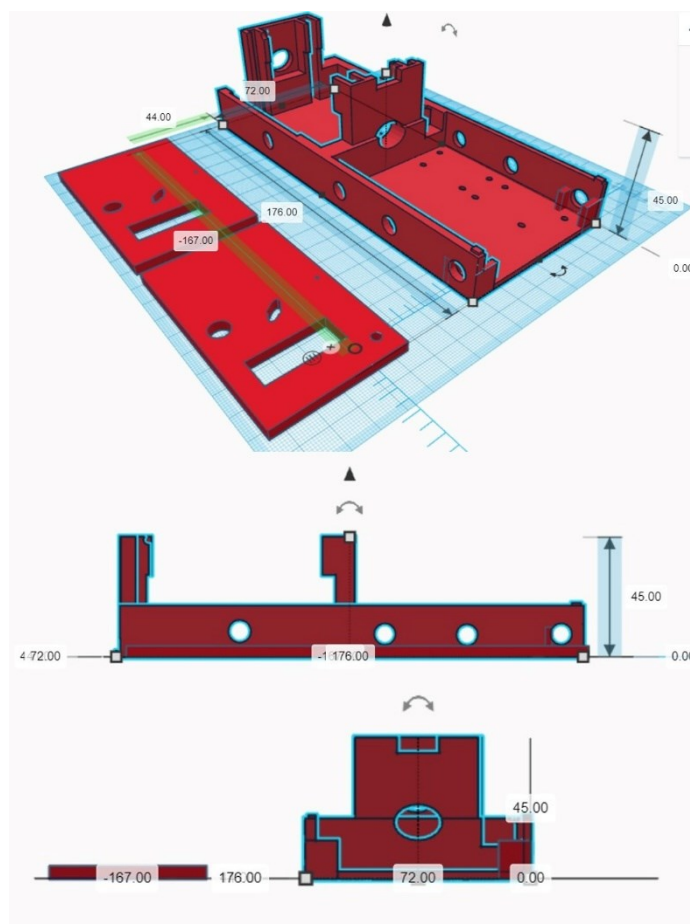
Pro návrh 3D modelu jsme použili volně dostupnou aplikaci Tinkercad, která svou přístupností a jednoduchým ovládním plně vyhovuje tvorbě modelu robotického auta pro 3D tisk. Před finální verzí robota jsme vytvořili několik prototypů. Jedním z počátečních rozhodnutí bylo umístění motorů a tedy, zda bude auto mít pohon na přední nebo zadní nápravu. Zvolili jsme pohon na přední nápravu, což je běžné u většiny aut. Robot se tedy pohybuje pomocí dvou krokových motorů, přičemž každý z nich pohání jedno kolo. Motory nejsou připevněny přímo ke kolům auta, ale k ozubeným kolům, která fungují jako

převodovka. Výstup motoru je připojen k většímu ozubenému kolu, které následně pohání menší ozubené kolo, a to roztáčí jednotlivá kola auta. K minimalizaci prokluzování kol jsme na přední kola přidělali gumu a přesunuli váhu z motoru blíže na přední část auta, aby kola neprokluzovala.

Pro napájení robota jsme využili lithium-polymerovou baterii GeB RC LiPol Baterii 2S 1000mAh 7.4V 20C s konektorem JST/BEC. Baterie byla zvolena z několika důvodů. Za prvé jako možnost externího napájení, které umožňuje, aby robot fungoval jako izolovaný objekt nezávislý na napájení z počítače. Za druhé, specifické energetické požadavky dvou krokových motorů, které robot využívá k pohybu, vyžadují vyšší napájení, jelikož USB z počítače není k pohybu obou motorů dostačující.

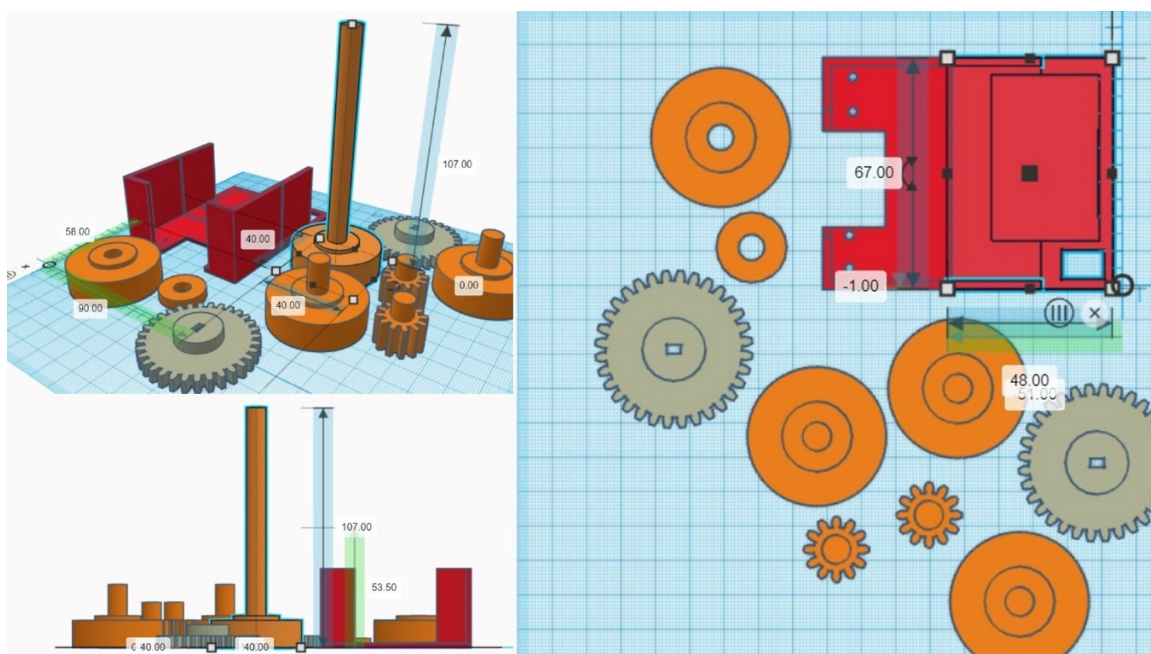
Největší výzvou při navrhování modelu bylo správné upevnění motorů s ozubenými koly, které slouží jako převody pro kola vozidla. Kola nesmějí být příliš volná, ale ani příliš blízko u sebe, aby se motory nezadrhávaly. Pokud by ozubená kola byla příliš daleko od sebe, nedotýkala by se, menší kolo by se neotáčelo a auto by se nepohybovalo. Tento problém vyžadoval přesné nastavení v řádu několika milimetrů, aby se našel ideální stav mezi těmito dvěma extrémů. Musíme počítat s tím, že tisk nemusí být vždy dokonalý, a proto může být nutné upravit jednotlivé výtisky pomocí vrtačky nebo pilníku. V jednom případě může být díra vytištěna na požadovanou velikost, zatímco v jiném může být větší.

Na následujících obrázcích jsou zobrazeny dva modely, protože se nevešly na jednu plochu. Vše je modelováno pomocí základních tvarů a booleovských operací a transformací. Na prvním obrázku vidíme podvozek auta, který je rozdělen na dvě části. Přední část je uzavřená a určena pro motory, zatímco zadní část je využívána k připevnění řídicích jednotek krokových motorů. Vedle podvozku jsou kryty, které kryjí krokové motory pomocí šroubů na místě. Otvory v těchto krytech umožňují průchod výstupům motorů, na které jsou připevněna ozubená kola. Na druhém obrázku je zobrazen kryt, který zakrývá řídicí jednotky motorů a zároveň je využita jako podložka pro desku Arduino UNO. Přední kola jsou vymodelována sama o sobě, ale zadní kolo má zároveň v sobě osu, ke které je pak jednoduše druhé kolo přilepeno a jsou pouze táhnuty vozem za sebou.



*Obr. 17 – 3D model podvozku Arduino auta*

Na obr. výše vidíme pohled na celý model, pohled zleva a v posledním pohled zezadu. Příloha bude poskytovat odkaz na modely, kde vše je vlastnoručně modelováno až na ozubená kola. Délka, šířka a výška podvozku je následující v tomto pořadí (176 mm, 72 mm, 45 mm). Boční stěny kryjící motory mají rozměry (80 mm, 50 mm, 3,8 mm). Díry, které vidíme z boku, mají průměr 10 mm a jsou tvořeny booleovskými operacemi v Tinkercadu přímo nástroj díra. Díry, vytvořené pro upevnění řídicí jednotky motoru mají průměr 3 mm.



Obr. 18 – 3D model kol a stříšky Arduino auta

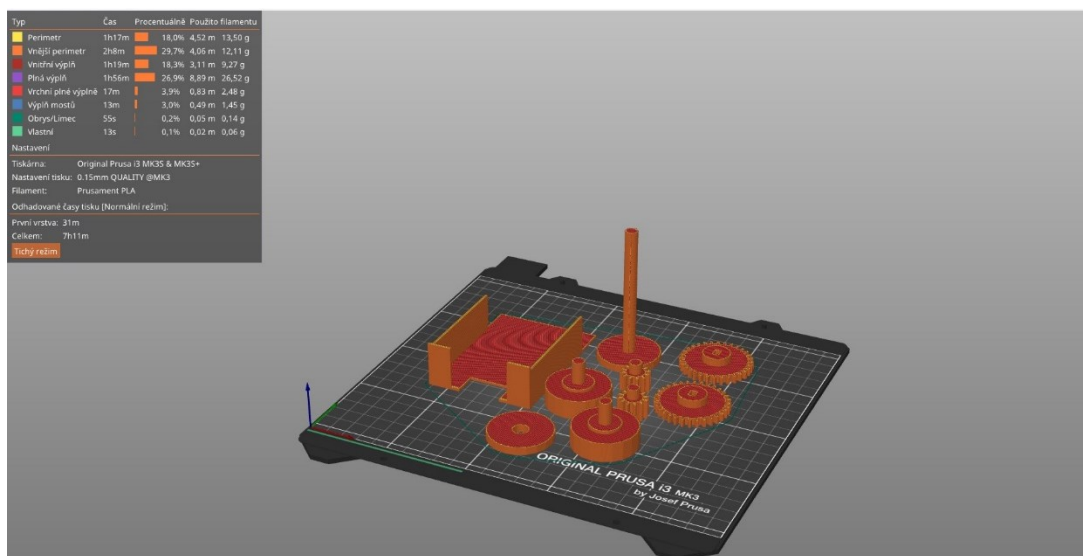
Průměry jednotlivých kol nabývají 40 mm, kde osa má tvar válce s průměrem 10 mm a výška i s osou je 28 mm. Ozubená kola jsou Royalty-free, jejich průměr je přibližně 45,25 mm a výška 10 mm. Nakonec stříška pro řídicí jednotku má délku, šířku a výšku (85 mm, 67 mm, 35 mm).

Pro kompletní popis chybí ještě zmínit baterii, nepájivé pole, tlačítko, senzory, odpory, kabely, čtyři fotorezistory a sonar na měření vzdálenosti. Proč je potřeba baterie, když můžeme Arduino napájet přes PC? Odpověď je jednoduchá: dva krokové motory by PC jednoduše neutáhlo, a navíc baterie poskytuje celkovou kompaktnost a nezávislost chytrého izolovaného objektu.

### 3.2.2 3D tisk, stavba a zapojení

Model byl vytištěn na 3D tiskárnách Prusa i3 MK3 s využitím aplikací PrusaSlicer a Pronterface. Jelikož tiskárna neumí přímo pracovat s formátem STL, tak musíme tento formát převést do podoby, které rozumí. K tomu využijeme program slicer. Slicer rozdělí model na jednotlivé vrstvy, s kterými již tiskárna dokáže pracovat, a následně je vytiskne. Kompletní robotické auto tiskneme ve dvou částech, protože kola a podvozek se nevejdou na jednu pracovní plochu, nebo tiskneme současně na dvou či více 3D tiskárnách. Pro žáky připravíme návod ke konstrukci robota, při které si procvičí práci s vrtačkou, pájkou

a šroubováky. Vrtačku použijeme k úpravě nedokonale vytištěných otvorů, pájku zase k připájení kabelů k zapínacímu/vypínacímu tlačítku. Na robota připevníme dvě malá nepájivá pole, na jedno z nich přivedeme + a, na druhé -. Žáci postaví robota podle návodu, který je součástí úlohy.



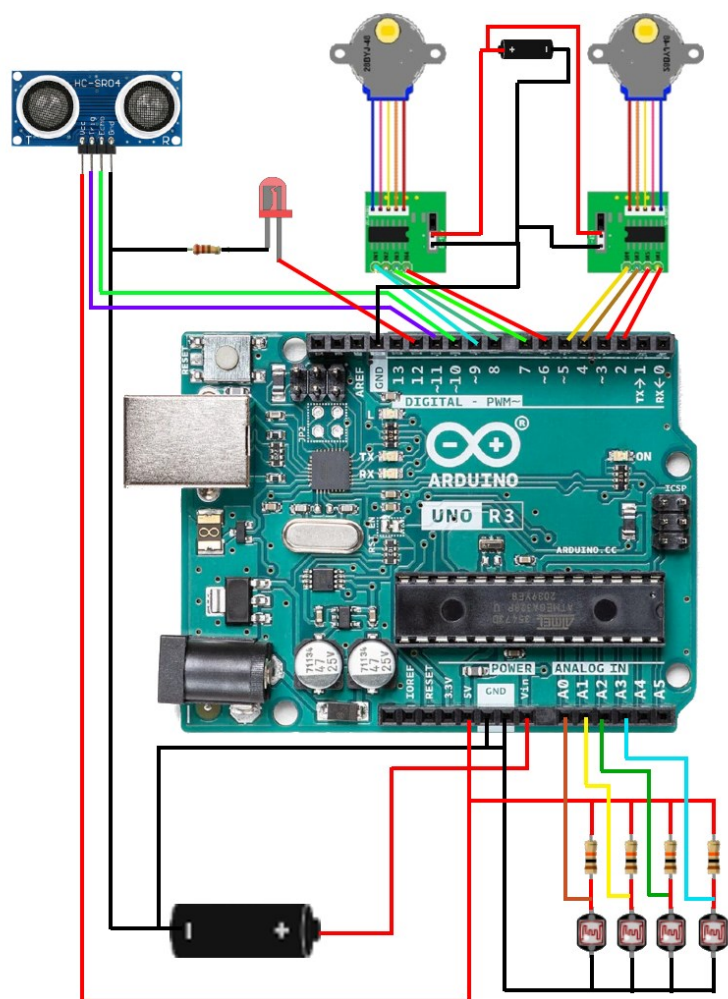
Obr. 19 – PrusaSlicer

Nyní se podíváme na důležité pojmy pro 3D tisk. Slicer jsme již zmiňovali, protože je klíčový nástroj pro tisk, převádí digitální 3D model na instrukce pro tiskárnu (G-code). Tyto instrukce určují pohyby tiskové hlavy, rychlost vytlačování materiálu a teplotu. Správné nastavení sliceru je zásadní pro kvalitu a pevnost tisku, přičemž uživatelé mohou upravovat parametry jako tloušťku vrstvy, rychlost tisku a teplotu. Extruder je srdcem tiskárny, které táhne a zahřívá filament do tekuté formy a vytlačuje jej tryskou. Perimetr označuje vnější část tisknutého objektu, která ovlivňuje jeho vzhled a pevnost. Infill označuje vnitřní vyplnění objektu, které může být vzorované pro úsporu materiálu. Support structures jsou dočasné podpurné pilíře pro převislé části objektu, které se po tisku odstraní. Materiál používaný pro samotný tisk se jmenuje filament, dostupný v různých typech s odlišnými vlastnostmi.<sup>76</sup>

76/ Klíčové termíny v 3D tisku a jejich význam. Online. 3D Stisk. 2024. Dostupné z: <https://3dstisk.cz/klicove-terminy-v-3d-tisku-a-jejich-vyznam/>. [cit. 2024-06-19].



Propojení jednotlivých součástek s Arduinem provedeme dle schématu (viz níže). Červené kabely budou značit 5 V a černé GND. Baterii připojíme k pinu VIN, což zajistí, že nedojde ke zkratu počítače nebo Arduina. Pokud bychom měli připojenou baterii a zároveň připojili Arduino k počítači pro nahrání kódu bez použití pinu VIN, ale pinu 5 V, došlo by ke zkratu. Následně připojíme jednotlivé součástky, jako jsou motory, sonar, fotorezistory a LED dioda. Každý motor má 4 cívky, což vyžaduje 4 digitální piny. Pro druhý motor platí to samé, dohromady tedy 8 digitálních pinů. Tím obsadíme digitální piny 2-9 pro ovládání krokových motorů. Pin 10-11 využijeme pro sonar Ttrig a Echo(I/O). Poslední obsazený digitální pin 12 pak použijeme pro LED diodu pro osvětlení ve tmě, kterou podpoříme rezistorem s odporem 220 ohmů. Nakonec obsadíme 4 analogové piny 0-3 fotorezistory, které vybavíme 10kilo-ohmovými rezistory.



Obr. 20 – Schéma zapojení Arduino a el. součástek robota

### 3.2.3 Programování

Program začíná deklarací proměnných, počínaje pomocnými proměnnými až po proměnné pro výpočty. Následuje deklarace čtyř cívek levého motoru a poté pravého motoru. Dále deklarujeme proměnné pro sonar (vstup/výstup), čtyři fotorezistory, LED diodu a pole hodnot 1 a 0 určující chod krokového motoru.

V setupu nastavíme pomocí funkce pinMode jednotlivé piny jako vstupní nebo výstupní. Program pokračuje vytvořením funkcí pro jízdu vpřed, vzad a pro otočku vlevo a vpravo. Jednotlivým cívkám přiřazujeme hodnoty HIGH (1) nebo LOW (0), abychom uvedli motory do pohybu. Těmito sekvencemi kroků rozpohybujeme motor, odtud také pochází název krokový motor. Jelikož jsou motory zrcadlově otočeny, jízdu dopředu nebo dozadu dosahujeme tím, že se jeden motor točí proti směru hodinových ručiček a druhý ve směru hodinových ručiček. Rotace je pak dosažena otáčením motorů stejným směrem. Varianty s polem od žáků nepožadujeme, důležitý je funkční kód.

Pohyb robota je naprogramován, zbývá přidat inteligenci pomocí senzoru vzdálenosti, který využíváme v další funkci k měření vzdálenosti. Dalšími senzory jsou fotorezistory, které měří intenzitu světla ve 4 směrech. Tyto funkce slouží ke sběru dat, která následně využíváme pro jízdu za světlem podle podmínek čtyř fotorezistorů a sonaru.

```
int j, i, k, l, vzdalenost_mm;
long mereni, vzdalenost;
int u1, u2, u3, u4;
int hranice_svetla = 500;

//Levy motor
int l1 = 2;
int l2 = 3;
int l3 = 4;
int l4 = 5;

//Pravy motor
int p1 = 6;
int p2 = 7;
int p3 = 8;
int p4 = 9;

//vzdalenost
int prijimac = 10; //echo INPUT
int vysilac = 11; // trig OUTPUT
```



```

// piny pro fotorezistory
int f1 = A0; // Zadní fotorezistor
int f2 = A1; // Levý z pohledu řidiče
int f3 = A2; // Pravý z pohledu řidiče
int f4 = A3; //Přední fotodioda

// pin pro světla
int s1 = 12;
//Pole motorů
int vpred[32] = { 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
1, 0, 0, 0, 1, 1, 0, 0, 1 };
int vzad[32] = { 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0,
0, 1, 0, 0, 1, 1, 0, 0 };

void setup() {
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(14, OUTPUT);
  pinMode(p1, OUTPUT);
  pinMode(p2, OUTPUT);
  pinMode(p3, OUTPUT);
  pinMode(p4, OUTPUT);
  pinMode(vysilac, OUTPUT);
  pinMode(prijimac, INPUT);
  pinMode(f1, INPUT);
  pinMode(f2, INPUT);
  pinMode(f3, INPUT);
  pinMode(f4, INPUT);
  pinMode(s1, OUTPUT);
}

void Otocka_Vpred() { //Zakovsky postup
  digitalWrite(11, 1); digitalWrite(12, 0); digitalWrite(13, 0); digitalWrite(14, 0); delay(1);
  digitalWrite(p1, 1); digitalWrite(p2, 0); digitalWrite(p3, 0); digitalWrite(p4, 0); delay(1);
  digitalWrite(11, 1); digitalWrite(12, 1); digitalWrite(13, 0); digitalWrite(14, 0); delay(1);
  digitalWrite(p1, 1); digitalWrite(p2, 0); digitalWrite(p3, 0); digitalWrite(p4, 1); delay(1);
  digitalWrite(11, 0); digitalWrite(12, 1); digitalWrite(13, 0); digitalWrite(14, 0); delay(1);
  digitalWrite(p1, 0); digitalWrite(p2, 0); digitalWrite(p3, 0); digitalWrite(p4, 1); delay(1);
  digitalWrite(11, 0); digitalWrite(12, 1); digitalWrite(13, 1); digitalWrite(14, 0); delay(1);
  digitalWrite(p1, 0); digitalWrite(p2, 0); digitalWrite(p3, 1); digitalWrite(p4, 1); delay(1);
  digitalWrite(11, 0); digitalWrite(12, 0); digitalWrite(13, 1); digitalWrite(14, 0); delay(1);
  digitalWrite(p1, 0); digitalWrite(p2, 0); digitalWrite(p3, 1); digitalWrite(p4, 0); delay(1);
  digitalWrite(11, 0); digitalWrite(12, 0); digitalWrite(13, 1); digitalWrite(14, 1); delay(1);
  digitalWrite(p1, 0); digitalWrite(p2, 1); digitalWrite(p3, 1); digitalWrite(p4, 0); delay(1);
  digitalWrite(11, 0); digitalWrite(12, 0); digitalWrite(13, 0); digitalWrite(14, 1); delay(1);
  digitalWrite(p1, 0); digitalWrite(p2, 1); digitalWrite(p3, 0); digitalWrite(p4, 0); delay(1);
  digitalWrite(11, 1); digitalWrite(12, 0); digitalWrite(13, 0); digitalWrite(14, 1); delay(1);
  digitalWrite(p1, 1); digitalWrite(p2, 1); digitalWrite(p3, 0); digitalWrite(p4, 0); delay(1);
}

```

```

void Otocka_Vpred_Pole() {
    int j, i;
    for (j = 0; j < 8; j++) {
        for (i = 0; i < 4; i++) {
            digitalWrite(l1 + i, vpred[j * 4 + i]);
            digitalWrite(p1 + i, vzad[j * 4 + i]);
        }
        delay(1);
    }
}

void Otocka_Vzad_Pole() {
    for (j = 0; j < 8; j++) {
        for (i = 0; i < 4; i++) {
            digitalWrite(l1 + i, vzad[j * 4 + i]);
            digitalWrite(p1 + i, vpred[j * 4 + i]);
        }
        delay(1);
    }
}

void Otocka_Vpravo_Pole() {
    for (j = 0; j < 8; j++) {
        for (i = 0; i < 4; i++) {
            digitalWrite(l1 + i, vzad[j * 4 + i]);
            digitalWrite(p1 + i, vzad[j * 4 + i]);
        }
        delay(1);
    }
}

void Otocka_Vlevo_Pole() {
    for (j = 0; j < 8; j++) {
        for (i = 0; i < 4; i++) {
            digitalWrite(l1 + i, vpred[j * 4 + i]);
            digitalWrite(p1 + i, vpred[j * 4 + i]);
        }
        delay(1);
    }
}

void Zmer_Vzdalenost() { //unčuje vzdálenost v milimetrech
    digitalWrite(vysilac, LOW);
    delayMicroseconds(20);
    digitalWrite(vysilac, HIGH);
    delayMicroseconds(30);
    digitalWrite(vysilac, LOW);
    delayMicroseconds(20);
    mereni = pulseIn(prijimac, HIGH);
    vzdalenost = mereni * 34 / 100;
}

```

```

void Zmer_Rezistory() {
    u1 = analogRead(f1); // zadní fotodioda
    u2 = analogRead(f2); // zadní fotodioda levá z pohledu řidiče
    u3 = analogRead(f3); // fotodioda pravá
    u4 = analogRead(f4); // přední fotodioda
    // Výpočet napětí by byl u1*5/1024;
}
void Jed_Za_Svetlem() {
    Zmer_Rezistory();
    if ((u1 < u2) && ((u1 < u3) && (u1 < u4))) { Otopka_Vpravo_Pole(); }
    if ((u2 < u1) && ((u2 < u3) && (u2 < u4))) { Otopka_Vlevo_Pole(); }
    if ((u3 < u1) && ((u3 < u2) && (u3 < u4))) { Otopka_Vpravo_Pole(); }
    if ((u4 < u1) && ((u4 < u2) && (u4 < u3))) { Otopka_Vpred_Pole(); }
}
void ujet_vzdalenost(int mm) { // Ujede zadanou vzdalenost v mm
    int otacek;
    int j = 0;
    otacek = (int)round(mm * 1.46);
    for (j = 0; j < otacek; j++) {
        otopka_vpřed();
    }
}
void otopka_uhel_vlevo(int mm) { //Otoci se o zadaný uhel vlevo
    int otacek;
    int j = 0;
    otacek = (int)round(mm * 2.06);
    for (j = 0; j < otacek; j++) {
        otopka_vlevo();
    }
}
void otopka_uhel_vpravo(int mm) { //Otoci se o zadaný uhel vpravo
    int otacek;
    int j = 0;
    otacek = (int)round(mm * 2.06);
    for (j = 0; j < otacek; j++) {
        otopka_vpravo();
    }
}

void loop() { // Jízda za světlem, není-li překážka
    Zmer_Vzdalenost();
    if (vzdalenost > 200) { Jed_Za_Svetlem(); }
    //}
}

```

### 3.3 Návrh námětů

K úlohám přistupujeme konstruktivisticky, proto jsme vybrali některé úlohy zaměřené pouze na rozvoj prekonceptů, které jsou pro tento typ výuky velmi důležité. Například v úloze s diódou žáci zkoumají Arduino desku, propojují elektronické součástky pro tvorbu obvodů, ale i představí programování, od deklarace proměnných po představení struktury programu (setup a loop), funkci digitalWrite, která na pin přiřazuje napětí buď 0, nebo 5 V. Máme tedy „prekonceptové“ úlohy, a také úlohy potřebné pro postup v konstrukci robota. V obou případech úloh je nezbytný učitelův výklad a komentář k důležitým znalostem, například k fungování krokového motoru. Přestože princip fungování motoru mohou vymyslet žáci, učitel poskytne základní informace, jako že se krokový motor skládá ze čtyř cívek, a ukáže obrázek, na kterém žáci odvodí algoritmus pro dosažení pohybu pomocí Arduina.

Úlohy zahrnují 3D modelování, tisk, stavbu robotů, propojení elektrických obvodů a programování. V těchto oblastech je často nutné provádět různé výpočty, takže rozvoj matematického myšlení se stává klíčovou zastřešující oblastí pro všechny tyto činnosti. Díky této kombinaci přichází v potaz varianta STEM výuky.

Ke vzdělávacím cílům řekněme, že se žáci naučí používat 3D modelovací software pro vytvoření návrhu robota a využívat geometrické koncepty k vytváření a úpravě 3D objektů. Seznámí se s přípravou modelů pro 3D tisk a obsluhou 3D tiskárny, čímž si rozvinou motorické schopnosti při stavbě robota. Pochopí funkčnost jednotlivých součástí robota a osvojí si základní práci s mikrokontrolerem Arduino, včetně zapojení motorů, senzorů a dalších komponent. V oblasti programování se seznámí se základními programovacími strukturami (podmínky, cykly a funkce) a naučí se programovat Arduino pro ovládání pohybu robota a interakci se senzory. Celý proces rozvíjí jejich logické a matematické myšlení. Projekt podporuje rozvoj klíčových kompetencí, jako jsou kompetence k učení, řešení problémů, komunikativní, sociální a personální, občanské a pracovní, díky kooperativní výuce a zvolenému tématu.

#### **ÚLOHA 1. Seznámení s aplikací Tinkercad**

Nauč se pracovat s prostředím Tinkercad, objev jednotlivé nástroje, seznám se s ovládáním pracovní plochy, s dostupnými objekty a jak s nimi pracovat.

### **Očekávané výstupy: Žák/žákyně**

- definuje základní pojmy a nástroje softwaru pro 3D modelování.
- vyzkouší různé techniky umístování a manipulace s objekty na pracovní ploše.
- demonstruje efektivní pohyb v prostoru a pracovní ploše.
- rozliší mezi jednotlivými typy objektů a jejich funkcemi.

### **Metodické poznámky:**

- Začněte diskusí o významu 3D modelování v různých oblastech (např. architektura, inženýrství, design).
- Dejte žákům čas, aby sami prozkoumali prostředí Tinkercadu. Povzbudte je, aby si sami vyzkoušeli přidávat objekty a manipulovat s nimi.
- Vysvětlete základní nástroje a pojmy na základě toho, na co se žáci dotazovali a co sami objevili.

### **ÚLOHA 2. Modelování základních tvarů**

Vymodeluj kvádrou nádrž, sud, pohár a konvičku.

### **Očekávané výstupy: Žák/žákyně**

- aplikuje jednotlivé nástroje a transformace k navržení jednotlivých objektů
- rozlišuje mezi geometrickými tvary a specifikuje, které jsou vhodnější pro tvorbu jednotlivých částí modelů.
- analyzuje efektivitu různých nástrojů pro modelování komplexních modelů.

### **Metodické poznámky:**

- Diskutujte s žáky o různých tvarech v reálném světě a jejich využití a o primitivních tvarech, z kterých se mohou skládat.
- Poskytněte tipy pro seskupování a zarovnání objektů.

### **ÚLOHA 3. Modelování dílů pro robota**

Vymodeluj kola a kryt pro robota a odhadni přibližné rozměry a následně ověř odhad změřením pomocí metru.

**Očekávané výstupy:** Žák/žákyně

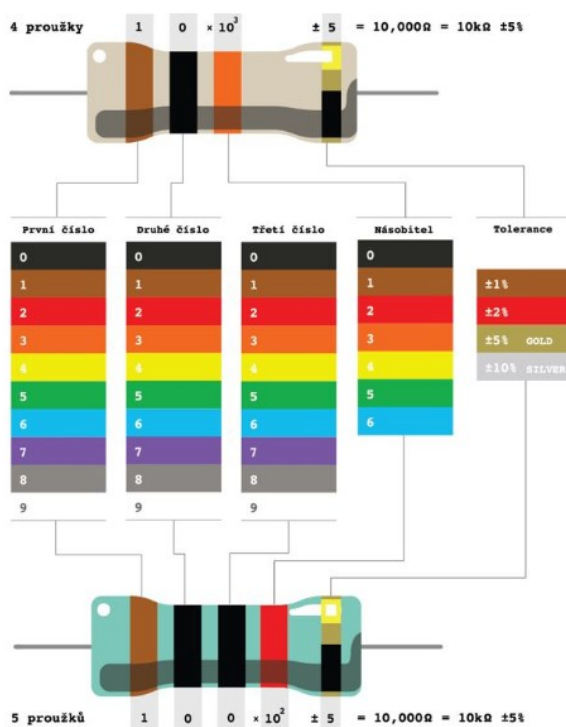
- navrhne model pomocí základních geometrických tvarů.
- aplikuje booleovské operace (funkci díra, sloučení) pro tvorbu modelu.
- analyzuje rozměry a proporce modelu pro měřítko robota.

**Metodické poznámky:**

- Nechte kolovat model robota, aby žáci mohli odhadnout velikost jednotlivých komponent.
- Nechte žáky reflektovat, co se naučili o používání základních geometrických tvarů a booleovských operací při modelování dílů pro robota.

#### **ÚLOHA 4. Určení velikosti odporů**

Podle ilustračního schématu urči hodnoty odporů jednotlivých rezistorů.



77

*Obr. 21 – Schéma pro určení hodnoty rezistoru*

<sup>77</sup>/ NOVÁK, Milan a PECH, Jirí. Robotika pro střední školy: programujeme Arduino. Jihočeská univerzita v Českých Budějovicích, 2018. ISBN 978-80-7394-786-6.

**Očekávané výstupy:** Žák/žákyně

- aplikuje znalosti o barvových kódech rezistorů k jejich správné identifikaci.
- rozliší jednotlivé rezistory do skupin a určí jejich hodnoty odporu.

**Metodické poznámky:**

- Ukažte žákům více rezistorů s různými hodnotami a nechte je zkoumat rozdíly mezi odpory.

### **ÚLOHA 5. Výpočet hodnoty odporu**

Vypočítej hodnotu odporu rezistoru v obvodu s Arduinem a LED diodou s napětím 5 V a proudem 2 mA. Urči, který rezistor ze skupiny vybrat. Popiš, jak se mění vztah mezi odporem a proudem při konstantním napětí 5 V a změně jedné z těchto veličin.

**Očekávané výstupy:** Žák/žákyně

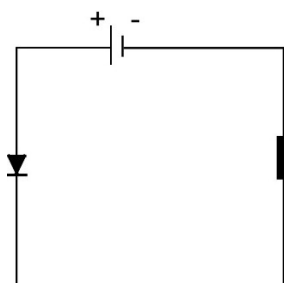
- identifikuje Ohmův zákon a vysvětlí tento vztah.
- použije Ohmův zákon pro výpočet odporu dle stanovených parametrů.
- analyzuje změnu ve vztahu při konstantním napětí 5 V a zvýšení jedné ze dvou veličin.

**Metodické poznámky:**

- Podle předešlé úlohy nechte žáky identifikovat správný rezistor.

### **ÚLOHA 6. Sestavení obvodu s LED diodou**

Sestav obvod s rezistorem a diodou podle schématu tak, aby se LED dioda rozsvítila. Využij nepájivé pole a desku Arduino.



*Obr. 22 – Schéma zapojení LED diody a rezistoru*

**Očekávané výstupy:** Žák/žákyně

- popíše desku Arduino a nepájivé pole.
- rozliší katodu a anodu u LED diody.
- demonstruje zapojení el. obvodu podle schématu.

**Metodické poznámky:**

- Nechte žáky prozkoumat komponenty, které budou používat.
- Proveďte praktickou ukázkou, jak rozpoznat katodu a anodu u LED diody.
- Otestujte v obvodu jiné odpory a jaký mají vliv.

### **ÚLOHA 7. Arduino a blikající dioda**

Sestav obvod s Arduino deskou a blikající diodou, naprogramuj ji tak, aby blikala v intervalu půl vteřiny.

**Očekávané výstupy:** Žák/žákyně

- používá Arduino IDE, seznámí se s prostředím
- zvolí správný typ desky a správný port
- rozliší mezi digitálním/analogovým pinem a vstupem/výstupem.
- vytvoří program, který na Arduino desku nahraje a otestuje jeho funkčnost.
- ovládá světelné výstupy.

**Metodické poznámky:**

- Začněte s žáky prozkoumáním Arduino IDE. Projděte si prostředí a jak propojit desku s PC.
- Tato úloha je zaměřena na získání základních dovedností v programování a seznámení se s Arduino příkazy, proto se můžeme přiklonit k transmisivnímu (instruktivnímu) přístupu.

### **ÚLOHA 8. Arduino a tři LED diody**

Sestav obvod s Arduino deskou a třemi LED diodami, naprogramuj je tak, aby postupně blikaly v intervalu jedné vteřiny.



**Očekávané výstupy:** Žák/žákyně

- dokáže analogicky sestavit obvod s vyšším počtem el. součástek.
- vyčíslí interval blikání převodem na milisekundy
- modifikuje původní program pro ovládání více el. součástek.

**Metodické poznámky:**

- Diskutujte o tom, jak se sestavuje obvod s více LED diodami a jaké jsou možnosti jejich zapojení.
- Upozorněte na využívání milisekund v Arduino.

### **ÚLOHA 9. Princip krokového motoru**

Prostuduj a vysvětli, jak funguje krokový motor.

**Očekávané výstupy:** Žák/žákyně

- pojmenuje základní části krokového motoru (např. stator, rotor).
- diskutuje o přenosu elektrických impulzů na mechanické pohyby.
- objasní jednotlivé kroky ke kompletní otočce hřídele.

**Metodické poznámky:**

- Promítněte obrázek krokového motoru na plátno a diskutujte o jeho principu fungování.
- Projděte s žáky jedno celé otočení.

### **ÚLOHA 10. Funkce pro pohyb krokového motoru**

Připoj krokový motor s řídicí jednotkou k Arduino a vytvoř funkci otoccka() pro pohyb krokového motoru, který má 4 cívky.

**Očekávané výstupy:** Žák/žákyně

- objasní zapojení krokového motoru s deskou Arduino.
- kombinuje znalosti o krokových motorech a programování Arduina k vytvoření funkce pro pohyb.
- diskutuje s žáky smysl funkcí v programování.

### **Metodické poznámky:**

- Zmiňte důvod potřeby využití řadiče pro zapojení motoru. Požadovaný proud pro ovládání cívky je příliš vysoký, řadič tento proud posílí.

### **ÚLOHA 11. Tisk robota**

Vytiskni robota s pomocí programů Prusa Slicer a PronterFace.

**Očekávané výstupy:** Žák/žákyně

- interpretuje využití jednotlivých programů a princip 3D tisku.
- vyzkouší výtisk modelu robota.

### **Metodické poznámky:**

- Vysvětlete žákům základní principy 3D tisku a rozdíly mezi těmito programy.
- Diskutujte možné obtíže při tisku a jak na ně reagovat.

### **ÚLOHA 12. Sestavení robota**

Sestav robota podle následujícího návodu. Proveď konstrukci a zapojení jednotlivých součástí s Arduinem.

*Pomůcky: vytištěné části robota, smirkový papír, sada vrtáků, vrtačka, šrouby, podložky, matičky, vteřinové lepidlo, šroubovák, Arduino, 2x krokový motorek, vypínač, pájka, stahovací pásek.*

*1. Připoj zkušebně velké ozubené kolo k motorku. Pilníčkem nebo vrtačkou případně dooprav otvor v kole.*

*2. Zasuň zkusmo motorové destičky do podvozku a případně je smirkovým papírem lehce opracuj. Zkontroluj, zda se přední kola lehce otáčejí v podvozku, pokud ne, pilníkem nebo vrtačkou otvor oprav. Totéž proved' s malým ozubeným kolem a motorovou destičkou. V motorové destičce vrtákem 3,5 zvětši otvory pro pásek na baterii.*

*3. Přišroubuj motorky bez ozubených kol k motorovým destičkám a nasuň na osu motorku velké ozubené kolo.*

4. Prostrč malé ozubené kolo do patřičného otvoru a zkontroluj dolehnutí zubů. Pokud je mezera mezi zuby příliš velká, posuň motorek, případně pooprav příslušné otvory na motorové destičce.
5. Na přední kola dej gumovou pneumatiku z duše a slep je s malým ozubeným kolem. Pozor na přilepení!
6. K vypínači připájej dva dostatečně dlouhé vodiče, vypínač vsuň do krytu robota a přilep jej.
7. Přišroubuj k podvozku desky plošných spojů od motorků a připoj k nim motorky.
8. Nad motorky připni stahovacím páskem baterii.
9. Na násadu na podvozek přišroubuj Arduino a přilep dvě nepájivá pole popiš je + nebo -.
10. Připoj Arduino k motorkům a napájení motorků a Arduina k nepájivým polím.
11. Izolepou zabezpeč spoje a spoj skupiny vodičů.
12. Uploaduj do Arduina sketch pro jízdu.
13. Zavolej učitele na kontrolu a pod jeho dohledem připoj záporný pól nabitě baterie, a nakonec kladný od vypínače.
14. Zapni vypínačem robota a sleduj, zda se deska rozsvítí.

**Očekávané výstupy:** Žák/žákyně

- identifikuje základní komponenty robota a jejich funkce.
- pro potřebné úkony používá nástroje, jako jsou pájka a vrtačka.
- sestaví robota podle návodu, včetně konstrukce a zapojení jednotlivých součástí.
- analyzuje správnost konstrukce a hledá potenciální chyby.

**Metodické poznámky:**

- Pojmenujte a představte jednotlivé součástky a nástroje potřebné pro stavbu robota.
- Upozorněte žáky na bezpečnost a možnost zkratu.

### **ÚLOHA 13. Principy funkcí v programování**

Vysvětli, jak fungují funkce v programování a jaké jsou jejich typy.

**Očekávané výstupy:** Žák/žákyně

- pojmenuje základní typy funkcí
- identifikuje klíčové prvky, jako jsou název, parametry, tělo, funkce a návratový typ.
- posoudí přínos funkcí pro celkovou strukturu a údržbu kódu.

**Metodické poznámky:**

- Nechte žáky vytvořit jednoduché funkce v programovacím jazyce, který používají. Např. funkci, která počítá součet dvou čísel.
- Zaměřte se na tvorbu funkcí a na proces jejich volání, aby žáci rozuměli rozdílu mezi nimi.

### **ÚLOHA 14. Vytvoření funkcí pro pohyb robota**

Naprogramuj funkce `jizda_vpřed()`, `jizda_vzad()`, `otocka_vpravo()` a `otocka_vlevo()` pro kompletní pohyb v prostoru.

**Očekávané výstupy:** Žák/žákyně

- analogicky deklaruje proměnné pro dva motory.
- opakuje postup programu pro jízdu vpřed.
- analyzuje funkčnost a správnost naprogramovaných funkcí pro řízení krokových motorů.
- řeší problém s otočkou motoru pozpátku.
- prokáže funkčnost jednotlivých funkcí.

**Metodické poznámky:**

- Nechte žáky tvořit funkci pro jízdu vpřed. Pravděpodobně vytvoří funkci `otocka()`, protože si neuvědomí, že jsou motory zrcadlově převráceny. Přichází optimální prostor na hledání chyb a analýzu problému.

### **ÚLOHA 15. Pohyby robota**

Změř vzdálenost, kterou robot ujede za 512 otoček, a naprogramuj funkci pro ujetí zadané vzdálenosti. Obdobně naprogramuj funkci pro otočení robota o požadovaný úhel.

**Očekávané výstupy:** Žák/žákyně

- provede pomocí pravítka a úhlooměru měření ujeté vzdálenosti a otočení robota za 512 otoček krokového motoru.
- využije vztahy pro přímou úměrnost pro výpočet otoček motoru potřebných k ujetí zadané vzdálenosti či úhlu k otočení.
- používá cyklus for v těle funkce s parametrem.

**Metodické poznámky:**

- Vysvětlíte úlohu a její důvod, předložíte pomůcky pro měření.
- Proved'te s žáky více měření a spočítejte průměr v excelu.

### **ÚLOHA 16. Určení rychlosti robota**

Stanov rychlost pohybu robota tím, že změříš čas potřebný k ujetí určité vzdálenosti.

**Očekávané výstupy:** Žák/žákyně

- změří rychlost robotovy jízdy.

**Metodické poznámky:**

- Pomocí naměřených hodnot času a ujeté vzdálenosti nechte žáky vypočítat rychlost pohybu robota.

### **ÚLOHA 17. Jízda slalomem**

Naprogramuj robota, aby projel slalomovou dráhu podle dohody celé třídy.

**Očekávané výstupy:** Žák/žákyně

- analyzuje trasu a využívá funkce k pohybu po dráze.
- navrhuje optimalizace a zlepšení kódu na základě zpětné vazby a výsledků testování.
- rozebírá, jak různé programovací strategie jiných skupin ovlivňují schopnost robota projet dráhu efektivněji.

**Metodické poznámky:**

- Uspořádejte soutěž o nejrychlejší průjezd trasou, která žáky nadchne.
- Diskutujte strategie skupin.

**ÚLOHA 18. Ultrazvukové čidlo**

Prostuduj princip fungování ultrazvukového čidla a jeho připojení k desce Arduino.

**Očekávané výstupy: Žák/žákyně**

- identifikuje hlavní součásti ultrazvukového čidla (vysílač, přijímač).
- vysvětlí, jak ultrazvukové čidlo měří vzdálenost pomocí vysílání a přijímání zvukových vln.

**Metodické poznámky:**

- Diskutujte o principu měření vzdálenosti pomocí ultrazvukových vln. Vysvětlete, jak vysílač vysílá ultrazvukové signály a přijímač zachytává odražené signály.
- Popište, jak propojit senzor s deskou Arduino.

**ÚLOHA 19. Měření vzdálenosti**

Osad' robota ultrazvukovým čidlem vzdálenosti, zapoj ho a vytvoř funkci pro měření vzdálenosti.

**Očekávané výstupy: Žák/žákyně**

- připojí do obvodu senzor a vytvoří program, který zpracuje informace ze senzoru.
- používá proměnné pro uchování a zpracování dat ze senzoru.
- používá knihovnu NewPing, podmínku a cyklus for.

**Metodické poznámky:**

- Diskutujte o různých typech senzorů a jejich využití v reálném životě (parkovací senzory v automobilu, senzory u robotického vysavače).
- Připravte si schéma zapojení sonaru.

## **ÚLOHA 20. Reakce na překážku**

Naprogramuj robota, aby reagoval na překážku zastavením, otočením a pokračováním v novém směru.

**Očekávané výstupy:** Žák/žákyně

- vytvoří funkci pro zastavení pohonu robota.
- řeší problém propojením Arduino desky se senzorem a vytváří program, který zpracovává informace ze senzorů a odesílá je k výstupům.
- používá podmínky a cyklus for.

**Metodické poznámky:**

- Diskutujte s žáky o potřebě a významu takové funkce pro bezpečný pohyb robota v prostoru.
- Instruuje žáky, jakým způsobem používat podmínky pro rozhodování robota na základě informací ze senzoru.

## **ÚLOHA 21. Princip fotorezistoru**

Prostuduj a vysvětli princip fotorezistoru.

**Očekávané výstupy:** Žák/žákyně

- vysvětlí princip rezistoru, závislost intenzity světla na odporu.

**Metodické poznámky:**

- Diskutujte s žáky o tom, jak se odpor fotorezistoru mění v závislosti na množství dopadajícího světla.
- Proveďte praktické experimenty, kde žáci budou měřit odpor fotorezistoru při různých intenzitách osvětlení.

## **ÚLOHA 22. Světla ve tmě**

Připoj fotorezistor a LED diodu k robotovi a vytvoř program, který rozsvítí diodu ve tmě.

**Očekávané výstupy: Žák/žákyně**

- identifikuje fotorezistor a LED diodu mezi součástkami.
- opakuje připojení LED diody a fotorezistoru k desce Arduino.
- používá proměnné pro uchování a zpracování dat ze senzoru.
- programuje Arduino pro čtení dat z fotorezistoru a ovládání LED diody na základě intenzity světla.

**Metodické poznámky:**

- Upozorněte na digitální a analogové vstupy.
- Směřujte žáky k používání proměnných pro uchování naměřených hodnot ze senzoru a podmínky pro rozhodování, kdy má být LED dioda zapnuta, nebo vypnuta.

**ÚLOHA 23. Jízda za světlem**

Připoj čtyři fotorezistory směřující do různých směrů k robotovi a naprogramuj ho, aby jel za světlem.

**Očekávané výstupy: Žák/žákyně**

- vybere čtyři fotorezistory a jejich správné umístění na robotovi, které propojí s Arduino deskou dle schématu.
- analyzuje data z jednotlivých fotorezistorů a určuje směr jízdy robota.
- kombinuje data z fotorezistorů do algoritmu, který umožňuje robotovi sledovat zdroj světla.

**Metodické poznámky:**

- Předved'te žákům správný způsob zapojení fotorezistorů, včetně doplnění o rezistory.



### 3.4 Charakteristika účastníků případové studie

Vzorkem respondentů pro tuto případovou studii je třída 9. ročníku ZŠ Na Lukách v Poličce. Výzkumu se zúčastnila jedna vybraná skupina, která zahrnovala 11 žáků ( $N = 11$ ) s vyšším zájmem o informatiku, ale na informatiku nebyla třída nijak specializovaná. Skupina se skládá ze 3 dívek a 8 chlapců, kteří zůstávají anonymní. Výuka probíhala od začátku ledna do června, a to jednou týdně po dobu jedné hodiny, přičemž tři týdny zahrnovaly dvouhodinovou výuku. Součástí výuky bylo také jednodenní programovací soustředění o víkendu na chalupě.

Výhodnou metodou pro robotiku je kooperativní výuka, proto byli žáci rozděleni do 5 skupin, ve kterých pracovali po celou dobu. Čtyři skupiny chlapců byly ve dvojicích a jedna skupina, složená z dívek, byla ve trojici. Ve skupinách si žáci automaticky rozdělili role a každý našel své místo, protože někteří vynikali v teoretických postupech, zatímco jiní byli manuálně zruční, což dohromady vytvořilo schopné týmy. Výuka jednotlivců by byla mnohem složitější jak z hlediska dostupnosti Arduino desek, tak z hlediska kreativity a schopnosti samostatně řešit úkoly.

Skupina prošla STEM výukou, kterou jsem v rámci této diplomové práce vedl já. Alternativně by jednotlivé předměty mohli vyučovat různí učitelé, kteří by spolupracovali na společně vytvořených úlohách pro celý robotický projekt. Celý proces byl poměrně časově náročný, a proto nikdo nestihl dokončit poslední úlohy s fotorezistory. Dvě skupiny však začaly pracovat se sonarem pro měření vzdálenosti, zatímco ostatní dokončili pohybové úlohy.

Vzorek žáků doposud neměl žádné zkušenosti s programováním ani 3D modelováním, takže byl v těchto oblastech prakticky nepopsaným listem. V ostatních oblastech však měli základní znalosti, například zapojování elektrických obvodů z fyziky a matematické dovednosti potřebné pro různé výpočty. Překvapilo mě, že spousta žáků byla při stavbě robota velmi zručně schopná od práce s vrtačkou, šroubovákem po práci s pájkou.

### 3.5 Sběr dat pro případovou studii a jejich analýza

Kvalitativní výzkum je zaměřen na ověření a vyhodnocení námětů úloh v rámci konceptu STEM v konstruktivisticky vedené výuce. Validitu výsledků kvalitativního výzkumu zajišťujeme prostřednictvím tzv. triangulace<sup>78</sup>. Tento pojem pochází z geodézie, kde k určení pozice nějakého bodu potřebujeme minimálně tři pevné body. Podobně v kvalitativním výzkumu, pokud chceme zajistit, aby nasbíraná data co nejvíce odpovídala realitě, musíme se opírat o více zdrojů informací. To znamená použití několika různých nástrojů pro sběr dat, která ověřují stejnou informaci.

Prvním a velmi zásadním nástrojem výzkumu je pozorování vzdělávacího procesu, tedy samotné ověření úloh ve výuce s žáky, které probíhalo přibližně půl roku. S celkovými výsledky jsem spokojen, avšak dovedu si představit, že by tento projekt mohl trvat i celý rok. Tato metoda nám umožňuje zjistit, zda jsou žáci schopni řešit námi navržené úlohy, a tím odpovědět na první hypotézu. Druhým nástrojem výzkumu jsou rozhovory s žáky, které mohou teoreticky poskytnout odpovědi na všechny tři hypotézy. Třetím nástrojem jsou pretesty a posttesty, které zkoumají dosažení očekávaných výstupů ve STEM předmětech. Tyto výstupy můžeme také sledovat prostřednictvím pozorování během hodin. Čtvrtým a posledním nástrojem je dotazník, ve kterém žáci hodnotí obtížnost jednotlivých úloh na Likertově škále (1–5). Tento nástroj je užitečný pro ověření druhé hypotézy, tedy identifikaci problémů a obtíží s úlohami. Výsledky z dotazníku můžeme porovnat s výsledky z pozorování a rozhovorů, kde jsme zaznamenali, co žákům činilo potíže.

#### 3.5.1 Pozorování

Žáci pracovali na robotickém projektu ve skupinách po dvou, s výjimkou jedné skupiny, která byla tvořena třemi členy. Skupinová práce byla zásadní pro úspěšnou realizaci projektu; individuální práce by byla buď nerealizovatelná, nebo by si vyžádala výrazně více času. Dělbá práce ve skupinách fungovala efektivně, protože v každé skupině byl jeden žák více řemeslně zručný, zatímco druhý disponoval schopností logicky uvažovat a řešit problémy. Z hlediska vzdělávacích cílů lze konstatovat, že každá skupina byla schopna

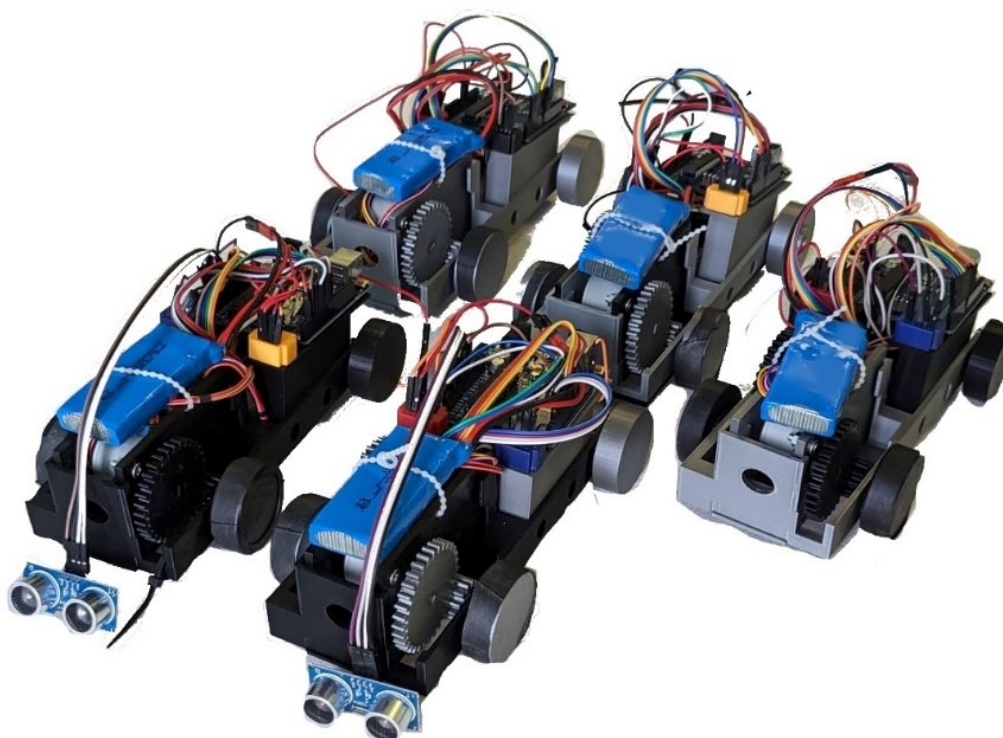
---

78/ Kvalitativní a kvantitativní výzkum, vzájemné porovnání. Online. Wikisofia. 2024. Dostupné z: [https://wikisofia.cz/wiki/3.\\_Kvalitativn%C3%AD\\_a\\_kvantitativn%C3%AD\\_v%C3%BDzkum,\\_vz%C3%A1jemn%C3%A9\\_porovn%C3%A1n%C3%AD](https://wikisofia.cz/wiki/3._Kvalitativn%C3%AD_a_kvantitativn%C3%AD_v%C3%BDzkum,_vz%C3%A1jemn%C3%A9_porovn%C3%A1n%C3%AD). [cit. 2024-06-21].

naplnit jak kognitivní, tak psychomotorické cíle. Kooperativní spolupráci byly navíc rozvíjeny i afektivní cíle, protože žáci se ve skupinách vzájemně chválili, podporovali a kriticky hodnotili své chyby.

Výuka byla vedena konstruktivisticky, přesto byly okamžiky, kdy bylo nezbytné provést přímý výklad nové látky. Konstruktivismus využívá prekoncepty, které žáci musí ovládat k efektivnímu řešení problémů. Z tohoto důvodu jsou na začátku úloh zařazeny „prekonceptové“ úlohy, které jsou vedeny konstruktivistickým přístupem. Vzhledem k tomu, že žáci dosud neprogramovali a neměli žádnou zkušenost s prostředím Arduino, bylo nutné jim poskytnout základní výklad o Arduinu, včetně popisu desky, jednotlivých příkazů a struktury programu. Úlohy 1–8 se zaměřují na rozvoj základních znalostí a dovedností pro práci s Arduinem, zapojování elektrických obvodů a elektronických součástek. Tyto úlohy nejsou přímo nezbytné pro stavbu robota, s výjimkou úlohy č. 3, kde se modeluje kryt a kola robota, a úloh č. 6 a č. 7, kde žáci zapojují a programují LED diodu. Tuto diodu může robot v závěrečné fázi využívat k osvětlení ve tmě. Úloha s LED diodou je klíčová „prekonceptová“ úloha, ve které žáci získají rozsáhlé znalosti a dovednosti, nezbytné pro další práci s Arduinem.

Obecně bych shrnul celý vyučovací proces a následně bych analyzoval jednotlivé oblasti a úlohy. Na obrázku níže je zobrazeno pět robotických vozidel, které žáci postavili. Dvě z těchto vozidel jsou vybavena senzory pro měření vzdálenosti, což umožňuje jejich plně autonomní pohyb a schopnost reagovat na okolí. Ostatní tři skupiny žáků úspěšně absolvovaly úlohy po úlohu č. 17, která zahrnovala průjezd slalomem. Také úspěšně splnili kritéria mého zadání, které požadovalo, aby jejich robot byl schopen pohybu ve všech směrech a bylo možné ho programovat. Například průmyslová robotická ramena v praxi plní také podobná kritéria, kdy se pohybují po přesně definované dráze a opakují stejný pohyb opakovaně.



*Obr. 23 – Žákovské výtvary robotických aut*

S výtvary žáků jsem spokojen. Některé výrobky se osvědčily více, některé méně, avšak všechny jsou funkční. Nejvíce problematické bylo pro žáky programování, následně zapojování elektronických součástek, matematické výpočty a fyzikální aspekty. Naopak nejjednodušší byla pro žáky fáze konstrukce, včetně pájení, vrtání, lepení, sestavování a navrhování jednotlivých součástek v aplikaci Tinkercad.

Navrhnutí robota od základů s žáky nebylo časově proveditelné, neboť jsem sám musel vytvořit několik prototypů, které jsem následně vytiskl a objevil několik dalších jiných problémů. Proto jsem předal žákům základní podvozek, ke kterému domodelovali kola a kryt. Dalo se očekávat, že programování bude hlavní výzvou, neboť žáci neměli žádné předchozí zkušenosti s programováním ani s používáním blokového programování.

Celkově všechny skupiny vybudovaly robotická vozidla na základě úloh až do úlohy č. 18. Všechny skupiny úspěšně absolvovaly slalom (úlohu č. 17) a seznámily se s principem fungování senzoru pro měření vzdálenosti. Dvě skupiny, které vykazovaly lepší výsledky, byly schopny dokončit úlohy č. 19 a 20. Zbylé tři skupiny paralelně dokončovaly funkce pro jízdu a slalom, které úspěšně zvládly, zatímco ostatní skupiny pracovaly na úlohách se

senzorem. K úlohám č. 21–23 jsme se s žáky nedostali, a tyto úlohy mohou sloužit jako budoucí návrhy pro další výuku a ověřování.

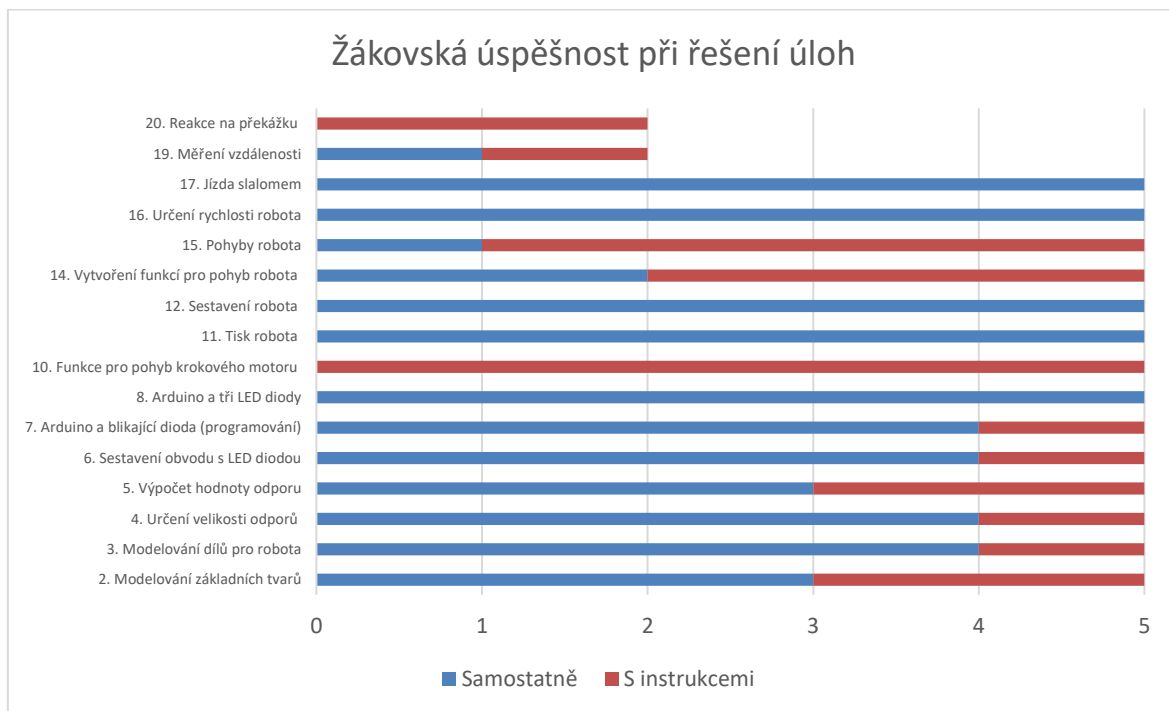
Domnívám se, že motivace žáků byla oproti standardním hodinám vyšší, protože mohli vidět hmatatelné výsledky své práce, což jim poskytovalo okamžitou zpětnou vazbu a posilovalo jejich porozumění. Tento praktický přístup je podle mě zajímavější než tradiční programování, kde se např. pouze vypisují výstupy na obrazovce, což může být více abstraktní.

V následující tabulce vidíme počet jednotlivých skupin, které vyřešily úlohy samostatně nebo s asistencí či instrukcemi. Některé úlohy byly vedeny spíše diskusí, proto nebylo možné přesně určit, kdo co zvládnul samostatně a co s asistencí, jelikož šlo o společnou diskusi. Data z úloh, ke kterým se žáci nedostali a pro které nejsou žádné výsledky, označuji (x).

Sběr dat z úloh žákovských řešení		
	Samostatně	S asistencí
1. Seznámení s aplikací Tinkercad	diskuse	diskuse
2. Modelování základních tvarů	3	2
3. Modelování dílů pro robota	4	1
4. Určení velikosti odporů	4	1
5. Výpočet hodnoty odporu	3	2
6. Sestavení obvodu s LED diodou	4	1
7. Arduino a blikající dioda	4	1
8. Arduino a tři LED diody	5	0
9. Princip krokového motoru	diskuse	diskuse
10. Funkce pro pohyb krokového motoru	0	5
11. Tisk robota	5	0
12. Sestavení robota	5	0
13. Principy funkcí v programování	diskuse	diskuse
14. Vytvoření funkcí pro pohyb robota	2	3
15. Pohyby robota	1	4
16. Určení rychlosti robota	5	0
17. Jízda slalomem	5	0
18. Ultrazvukové čidlo	diskuse	diskuse
19. Měření vzdálenosti	1	1
20. Reakce na překážku	0	2
21. Princip fotorezistoru	x	x
22. Světla ve tmě	x	x
23. Jízda za světlem	x	x

Tabulka 1 – Výsledky z úloh z pohledu samostatnosti při řešení

Průměrný počet úloh, které žáci vyřešili samostatně, je 0,6357, s asistencí 0,2875 a pro nesplněné úlohy 0,075. Pro náš vzorek 16 úloh pro 5 skupin, tedy 80 úloh. Kde 51 bylo vyřešených samostatně, 23 s asistencí a 6 nebylo vyřešeno.



*Graf 1 – Žákovská úspěšnost řešení úloh*

Nyní se zaměříme na analýzu jednotlivých vzdělávacích oblastí a na specifické obtíže, se kterými jsme se při řešení setkali. Začneme 3D modelováním, které rozvíjí žákovu prostorovou představivost a je z oblasti (Science a Technology). Při modelování, tedy **úlohy č. 1–3**, nebyly zásadní problémy; z tabulky vyplývá, že sice ne všichni žáci zvládli práci samostatně, ale obtíže byly spíše technického rázu, například nesprávné zarovnání jednotlivých objektů nebo chyby při výpočtu hran menšího kvádra pro vytvoření otvoru ve větším kvádra. Žáci již měli předchozí zkušenosti s 3D modelováním. Soustředil jsem se na ověření, zda žáci mají omezenou mírovou představivost. Zjistilo se, že žáci si nedokázali přesně představit velikost kola robota v poměru k celému robotu. Průměr kol byl 4 cm, ale jejich odhady velikosti byly často dvakrát až třikrát vyšší. Sami si pak měřením ověřili své odhady. Žáci intuitivně rozkládali jednotlivé objekty do základních primitivních tvarů, které následně kombinovali.

V následujícím bloku byly testovány znalosti z fyziky (Science, Engineering), konkrétně **úlohy č. 4–8**. Úloha zaměřená na určování odporu podle schématu odhalila problémy žáků při čtení barevných kódů rezistorů. Mnozí žáci četli barvy z opačné strany a nepostupovali správným směrem, jakým se mají barvy odporu číst. Dalším častým nedostatkem bylo nepochopení násobitele. Žáci sice správně určili pruhy před násobitelem, ale chybně je násobili nesprávnou konstantou místo mocninou deseti. Úloha č. 5, která by neměla být příliš kognitivně náročná, neboť se jedná o pouhé dosazení do vzorce, je přesto velmi důležitá. Tato úloha se konkrétně zaměřuje na určení odporu v našem el. obvodu. Úspěšně ji vyřešily pouze tři skupiny. Ostatní si nepamatovaly vzorec pro výpočet, nebo měly problémy s převody jednotek a výpočet provedly chybně. V úloze č. 6, která prověřuje znalosti zapojení dle LED diody a rezistoru s nepájivým polem a Arduino deskou, se žáci setkali s několika problémy. Někteří zaměnili kladný a záporný pól na nepájivém poli, což vedlo k nesprávnému zapojení neodpovídajícímu schématu. Další častou chybou bylo nesprávné umístění komponent do nepájivého pole, konkrétně do zdírek, které nebyly pod proudem. Někteří žáci neznali rozdíl mezi anodou a katodou LED diody, což vedlo k chybnému zapojení diody. Rovněž se projevila nejistota při práci s Arduinem, zejména ohledně správného připojení ke třem GND výstupům. Tyto chyby ukazují na potřebu lepšího porozumění základům práce s nepájivým polem a Arduinem a důkladnějšího procvičení praktických dovedností, aby se předešlo podobným chybám v budoucnu. Úloha č. 7 byla vysvětlena společně transmisivní výukou, protože žáci neznali příkazy jako DigitalWrite a deklaraci proměnných, což z ní činilo „prekonceptovou“ úlohu zaměřenou na základní příkazy Arduina. Pouze jedna skupina nedokázala správně zapojit obvod a nepochopila princip algoritmu blikání. Teoretickou chybou může být opět převod na milisekundy. Tato úloha je důležitá pro zvládnutí základních příkazů Arduina. Dále úlohu č. 8 zvládli postupně všichni žáci. Jedna skupina měla problém s deklarací proměnných a omylem zkopírovala stejnou proměnnou se stejným názvem vícekrát, což vedlo k nesprávnému definování tří LED diod.

Dalším kritickým blokem jsou **úlohy č. 9 a 10** (Technology, Engineering) zaměřené na objevování funkce krokového motoru a naprogramování jeho jedné otočky. Devátou úlohu jsme řešili diskusí, během níž jsme společně s žáky navrhli algoritmus pro ovládání krokového motoru a způsob, jak pouštět proud do jednotlivých cívek. Přesto žádná skupina

nebyla schopna převést tento algoritmus do programovacího jazyka. Posunuli jsme se dále až potom, co jsem napsal první řádek funkce, tedy prvního kroku krokového motoru. Tato úloha tedy vyžaduje revizi, protože žádný žák ji nebyl schopen samostatně vyřešit, kvůli nedostatečným zkušenostem s programováním a neznalostí syntaxe. Tím, že nehodnotíme schopnost žáků řešit úlohu zcela samostatně, hypotézu nezamítáme, ale musíme podotknout, že příkazy z úloh z LED diody by měly být dostačující pro její vyřešení. Shledávám problém i v časové dotaci, že žáci mohli jednoduše zapomenout příkazy, jelikož jsme se vídali jednou týdně a diody se programovaly před několika týdny, díky svátkům. Konstruktivistická výuka byla narušena, ale role učitele jako rádce může žákovi neschopnost odstranit, a lze se opřít o teorii zóny nejbližšího vývoje, kde žák využívá znalosti učitele k překonání obtíží. Proto bych hypotézu H1 prozatím nezamítnul. Byl jsem překvapen miskonceptem ohledně vstupu, kdy všechny skupiny nastavily cívky motoru jako vstupy, přestože by měly být nastaveny jako výstupy. Tento omyl může vzniknout z toho, že žáci cívky motoru konfigurovali jako vstupy a nastavovali jim hodnoty HIGH a LOW, což může vést k nesprávné interpretaci jejich funkce jako vstupů. Níže revidovaná úloha č. 10.

### **ÚLOHA 10. Funkce pro pohyb krokového motoru**

Připoj krokový motor s řídicí jednotkou k Arduinu a vytvoř funkci `otocka()` pro pohyb krokového motoru, který má 4 cívky, analogicky dle prvního kroku motoru.

```
digitalWrite(11, 1); digitalWrite(12, 0); digitalWrite(13, 0); digitalWrite(14, 0); delay(1);
```

Pro oblast dílen a rozvoje řemeslnictví jsou klíčové **úlohy č. 11 a 12**, tedy tisk a stavba robota (Technology, Engineering). Při tisku došlo v jednom případě k chybě tiskárny, avšak po jejím vyřešení proběhl proces bez dalších problémů, přičemž parametry pro tisk byly již předem nastaveny. Následně žáci měli k dispozici vytištěné součástky, které bylo nutné použít při konstrukci robota. Při samotném sestavování robota se však žáci potýkali s řadou technických problémů a chyb. V průběhu práce se opakovaně vyskytovaly problémy s pájením, kdy některé dráty odpadávaly nebo došlo k nekvalitnímu spojení. Jedna skupina dokonce při pájení zapojila tak nešikovně, že se přes kovové spoje roztavil plast, což mělo za následek nefunkčnost vypínače. Další problémy nastaly při vrtání děr, kdy některé skupiny provrtaly díry příliš velké, což vedlo k nedostatečnému doléhání ozubených kol. Při montáži ozubených kol došlo k další chybě, kdy si žáci omylem slepili ozubená kola



dohromady během lepení. Byl jsem příjemně překvapen jejich zručností a schopnostmi, zejména v manipulaci s vrtačkou, která pro ně nebyla ničím novým, a zaskočen prací s pájkou, jelikož již všichni ve škole dříve pájeli.

**Úlohy č. 13–17** se zaměřují na programování s využitím matematických výpočtů a jsou součástí technologického a matematického vzdělávání (Technology, Mathematics). Ihned bylo zřejmé, že tyto úlohy na programování jsou pro žáky náročné. V úloze č. 13 jsme společně s třídou probírali principy a diskutovali o fungování krokového motoru, přičemž jsme kreslili schematické obrázky s popisky pro následnou deklaraci. Úloha č. 14 představuje klíčový moment celého projektu, kde se žáci učí programovat různé typy pohybů robota: jízdu vpřed, dozadu, otočky doprava a doleva. Zajímavým aspektem této úlohy je, že žáci často předpokládají, že programují jízdu vpřed, avšak v praxi se jim objeví otočka, což vyžaduje jejich schopnost analyzovat situaci a identifikovat odlišnosti v chování robota. Mezi časté chyby žáků patřilo chybné pochopení označení cívek motorů. I přes nakreslený obrázek jednoho motoru s označením cívek L1, L2, L3 a L4 si žáci vytvářeli nová označení L5 až L8 a pracovali s proměnnými neuspořádaně. Nepochopili, že tato označení odpovídají jednotlivým cívkám levého a pravého motoru. Při objevení otáčky robota žáci identifikovali, že motory jsou zrcadlově otočeny, avšak měli potíže s vytvořením algoritmu pro otáčku motorů proti směru hodinových ručiček, nebo v některých případech dokonce problém nepoznali. Tato situace ukázala na potřebu hlubšího porozumění principům a správnému označení komponent při programování. Úloha č. 15 představovala pro žáky výraznou výzvu. Časté chyby se projevovaly v nesprávném výpočtu trojčlenky. Žáci měli za úkol spočítat, kolik otáček je potřeba pro určitou vzdálenost, avšak často špatně určili neznámou k dosažení specifikované vzdálenosti. Druhým zásadním problémem bylo nepochopení cyklu for, který byl pro žáky novým konceptem. Namísto samostatného řešení se spíše jednalo o diskusi, přičemž někteří žáci zcela nechápali jeho principy a použití. Konečně, žáci se potýkali s obtížemi v použití závorek, často přidávali nadbytečné závorky nebo nedokázali správně párovat závorky, což ovlivnilo nesprávnost jejich kódu. Tato úloha ukázala na potřebu dalšího zdokonalování v oblasti matematických výpočtů, cyklu for a správné syntaxe programování. Po úlohách, které nebyly žáky úplně nadšeně přijaty, se naskytly lépe přijaté úlohy, zejména úloha č. 16, kterou žáci úspěšně zvládli. Avšak někteří na začátku volili neoptimální vzdálenost kolem 30 cm, což vedlo k nepřesnému měření. Naopak úloha

č. 17 byla pro žáky nejzábavnější. V této úloze se věnovali hledání optimálního způsobu jízdy slalomem, přičemž si každý žák úspěšně osvojil volání funkcí a prostřednictvím diskuse hledali nejlepší strategii pro úspěšné dokončení úkolu.

Pro oblast senzorů zaměřujících se na **úlohy č. 18–20** jsme získali data pouze od dvou skupin, které dosáhly lepších výsledků a pracovaly obecně rychleji než zbývající tři skupiny. Úloha č. 18 byla realizována standardní diskusí jako nové téma, které se zaměřovalo na principy měření vzdálenosti. Jedna skupina se inspirovala řešením nalezeným na internetu, zatímco druhá skupina využila jejich přístup, řekněme tedy, že úlohu samostatně zvládla jedna skupina. U úlohy č. 20 všichni skončili, kde robot úspěšně dokázal zastavit podle překážky, ale nebyl schopen se následně otočit a pokračovat v jízdě. Poté už nebyl čas a věnoval se čas rozhovorům, testu a dotazníku.

**H<sub>1</sub>:** Žáci 9. ročníku ZŠ budou schopni sestavit a naprogramovat robotický systém s využitím Arduina a disponují dostatečnými znalostmi a dovednostmi řešit typ úkolů, které jsou součástí řešitelem připravené sady úloh.

Žáci prošli jednotlivými úlohami a vytvořili robotický systém, který je schopen být ovládán k určitému pohybu. Dvě skupiny z pěti vytvořily autonomní robotické vozidlo, které dokáže reagovat na překážky.

Na základě těchto výukových metod a nasbíraných dat je možné hypotézu H<sub>1</sub> přijmout. Neboli žáci 9. ročníku ZŠ jsou schopni sestavit a naprogramovat robotický systém s využitím Arduina a disponují dostatečnými znalostmi a dovednostmi řešit typ úkolů, které jsou součástí řešitelem připravené sady úloh.

### 3.5.2 Rozhovory

Dalším nástrojem pro sběr dat jsou rozhovory s žáky, které budou prezentovány ve formě parafrází jejich myšlenek a informací, které poskytli. Tyto rozhovory se zaměří na jednotlivé úlohy, ale také na obecnou problematiku spojenou s projektem a na zpětnou vazbu k němu. Počet osob zapojených do rozhovorů bude variabilní; někdy se bude jednat o celou třídu, jindy o skupinu žáků nebo jednotlivce. Rozhovory probíhaly během řešení jednotlivých úloh i po absolvování projektu.

#### **Blokové programování?**

Zajímavým podnětem byla zpětná vazba od čtyř žáků W, X, Y a Z, kteří se zúčastnili projektového dne na gymnáziu, kde se seznámili s variantou blokového programování Arduino, které dosud neznali. Vyučující je chválil za reagování na otázky a byl překvapen, že již mají s Arduinem zkušenosti. Žáci poznamenali, že blokové programování bylo mnohem jednodušší na pochopení než psaní celého kódu v IDE, jak to dělali ve škole. S tímto názorem se plně ztotožňuji, neboť je to hlavní překážkou žákova průchodu projektem. Neshledal jsem problém v obtížnosti úloh z pohledu algoritmizace nebo dílčích výpočtů, ale zásadním problémem byla samotná syntaxe programovacího jazyka a jeho struktura.

Žáci se s programováním sice poprali, ale pro usnadnění jejich práce by byla vhodná forma blokového programování, což sami žáci uvítali. Pozorování z hodin potvrdilo mé domněnky, žáci měli obrovský problém s uzávorkováním, které by blokové programování teoreticky odstranilo, protože jednotlivá vnořování jsou vizuálně lépe pochopitelná a viditelná. Náš pokus se zaměřoval na projekt, který by mohl být spíše na střední škole, proto jsem volil možnost psaní kódu. Závěrem lze konstatovat, že pokud by již žáci měli nějaké zkušenosti s programováním, přicházela by varianta psaní celého kódu v úvahu. Vzhledem k tomu, že tyto zkušenosti neměli, byla by volba blokového programování vhodnější.

V rozhovoru s každým z 11 studentů na otázku: Co bylo na celém projektu pro tebe nejobtížnější? nebylo snadné získat jednoznačné odpovědi, protože studenti měli problémy s vyjadřováním konkrétních obtíží. Jednotlivé výpovědi studentů se nakonec shodovaly v tom, že největší výzvou bylo programování. Nejvíce problémů jim způsobovalo správné používání závorek, zejména v situacích, kdy měli vnořené cykly uvnitř funkcí nebo v hlavní

smyčce programu (loop). Studenti často ztráceli přehled o tom, která závorka patří k jakému bloku kódu, což vedlo k chybám a komplikacím při ladění programu. Což zdůrazňuje důležitost správné struktury kódu a její srozumitelnost, kterou jsem opomíjel a spíš se soustředil na správnost kódu.

### **Rozhovory při řešení**

Uvedená situace se zaměřuje spíše na technické aspekty úloze č. 8, které je potřeba brát v potaz. Žáci úspěšně sestavili obvody s LED diodami a jejich programový kód byl také správně implementován, přesto se nic s diodami nedělo. Žáky jsem nasměroval ke kontrole zapojení vodičů, což se ukázalo jako hlavní zdroj potíží. Během pozorování jsem si všiml nízkých časových prodlev, a tak jsem se zeptal skupiny žáků „jestli jsou si vědomi, že časové prodlevy se nastavují v milisekundách?“, což nebrali v potaz. Následně jsme diskutovali o různých časových intervalech a jejich vlivu na viditelnost světelných efektů LED diod.

Při řešení úloh jsem se zeptal žáků  $x$  a  $y$ , proč při deklaraci programu označili cívky krokového motoru jako vstupy. Žák  $x$  odpověděl „že je to zřejmé, protože cívce přiřazují hodnoty HIGH nebo LOW, proto to musí být vstup“ a žák  $y$  ho doplnil „já jsem si myslel, že dát cívky jako vstupy je normální, protože jsme chtěli, aby mohly přijímat informace o tom, co se s nimi děje“. Zvláštností bylo, že si celá třída vytvořila podobný miskoncept.

Při řešení úlohy č. 10, týkající se programování krokového motoru s Arduinem, jsem vedl rozhovor s několika žáky, kteří se snažili vytvořit funkci `otocka()` pro pohyb krokového motoru se čtyřmi cívkami. Konkrétně se jednalo o situaci již po ukázce části kódu, jelikož žáci nebyli ničeho schopni. Žák  $x$  řekl: „Chápali jsme, jak algoritmus pro pohyb krokového motoru funguje. Věděli jsme, že musíme postupně aktivovat jednotlivé cívky v určitém pořadí. Dále celkově uvedli, že je nenapadlo použít funkci `digitalWrite()` k ovládní jednotlivých cívek. „Mysleli jsme, že to můžeme používat jenom u těch diod,“ přiznal žák  $x$ . Proto by se hodilo věnovat více času programování, chtěl jsem vyzkoušet s žáky jiné el. součástky jako je buzzer atd., ale nebyl čas. To by vedlo k tomu, že žáci nebudou myslet v izolovaných modelech (příkaz jenom pro diodu), ale obecně.

Za zmínku stojí rozhovor se skupinou, která se potýkala s programováním krokového motoru v úloze č. 14 pro opačný směr otáčení. Ukázalo se, jak důležitou roli hraje vizualizace

v porozumění danému technickému problému. Skupina byla pověřena úkolem vytvořit funkci pro řízení motoru proti směru otáčení. Programování chodu motoru ve směru probíhalo několik týdnů zpátky a žáci zapomněli princip. Nechal jsem žáky nakreslit obrázek a vizualizovat postup pomocí obrázku krokového motoru, který jim pomohl lépe porozumět fyzickému pohybu motoru. Po nakreslení motoru a připomenutí jeho principu pohybu byli schopni analogicky vytvořit kód pro otáčení motoru proti směru bez dalších obtíží. Pěkná byla spolupráce, kde jeden žák popisoval stavy cívek a druhý psal kód.

Při řešení úlohy č. 15 „pohyby robota“ týkající se přímé úměrnosti jsem vedl rozhovor s žákem  $x$ , který využíval povrchovou strategii řešení. Zeptal jsem se ho, jak přišel na svůj výsledek. Odpověděl: „Použil jsem trojčlenku, protože to tak děláme vždycky. Prostě jsem napsal tři čísla pod sebe a pak to nějak vypočítal.“ Když jsem se ho zeptal, jak vytvořil rovnici, přiznal: „Vlastně nevím, jak přesně to funguje. Jenom si zapíšu čísla a násobím nebo dělím podle potřeby. Rovnici bych asi nevytvořil, protože tomu úplně nerozumím.“ Tato výpověď ukazuje na nedostatečné pochopení principu přímé úměrnosti a potřeby hlubšího vysvětlení a procvičování.

### 3.5.3 Test

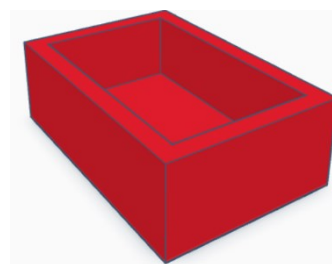
Test je rozdělen na oblasti matematika, fyzika a informatika a jeho cílem je zjistit zlepšení nebo zhoršení výsledků žáků v jednotlivých oblastech. Pretest byl psán žáky před realizací výuky robotického projektu a posttest po jejím ukončení. Nasbíraná data z těchto testů budou využita k přijetí, nebo zamítnutí hypotézy  $H_3$ .

Aby bylo možné data porovnávat, musí být testy stejné a nesmějí obsahovat přímo úlohy a otázky týkající se Arduina, protože žáci tuto oblast před výukou nemohli znát. Testy by měly testovat žákovo naplnění očekávaných výstupů z jednotlivých oblastí. Maximální počet bodů v testu je 13, přičemž každá z 13 úloh může být ohodnocena maximálně jedním bodem. Některé úlohy jsem hodnotil půlbodově za částečnou správnost. Test se skládá z následujících oblastí: matematika (4 úlohy), fyzika (5 úloh) a informatika (4 úlohy).

## Test

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?
2. Pokud se motor robotického auta otočí 512krát, celý robot změní svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.
4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr tak, aby vznikla vana s šířkou stěny 2 cm.



Obr. 24 – Test, 3D modelování

### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10\text{k}\Omega$  a vstupní napětí je 5 V?
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?
7. Jak poznáš, jak velký má rezistor odpor?
8. Vysvětli fungování fotorezistoru.



### 3.5.4 Analýza dat z testů

Analýzu provedeme nejprve pro celkové výsledky a následně pro jednotlivé oblasti: matematiku, fyziku a informatiku. Tím zjistíme, zda se žáci teoreticky zlepšili, a identifikujeme konkrétní oblasti, ve kterých ke zlepšení došlo.

Pretest x Posttest			
Žáci	Celkem pretest	Celkem posttest	Rozdíl v testech
1	2	6	4
2	2	7,5	5,5
3	1,5	2	0,5
4	3	3,5	0,5
5	3	6	3
6	4	10	6
7	5	11,5	6,5
8	3	7	4
9	2	1,5	-0,5
10	3,5	7	3,5
11	4,5	8	3,5
Průměr ( $\bar{x}$ )	3,045	6,364	-
Medián	3	7	-
Rozptyl ( $s^2$ )	1,157	8,686	-
Směr. Odchylka ( $s$ )	1,076	2,947	-

Tabulka 2 – Celkové výsledky z pretestu a posttestu

Z těchto výsledků je vidět, že průměrné skóre a medián se po projektu výrazně zvýšily. Rozptyl a směrodatná odchylka se také zvýšily, což může indikovat větší variabilitu výsledků v post-testu. Výsledky žáků před projektem jsou výrazně nižší a kompaktnější.

K ověření, zda došlo k významnému zlepšení výsledků studentů mezi pretestem a posttestem, vyžijeme Studentův t-test (párový). Formulujme hypotézu pro toto ověření:

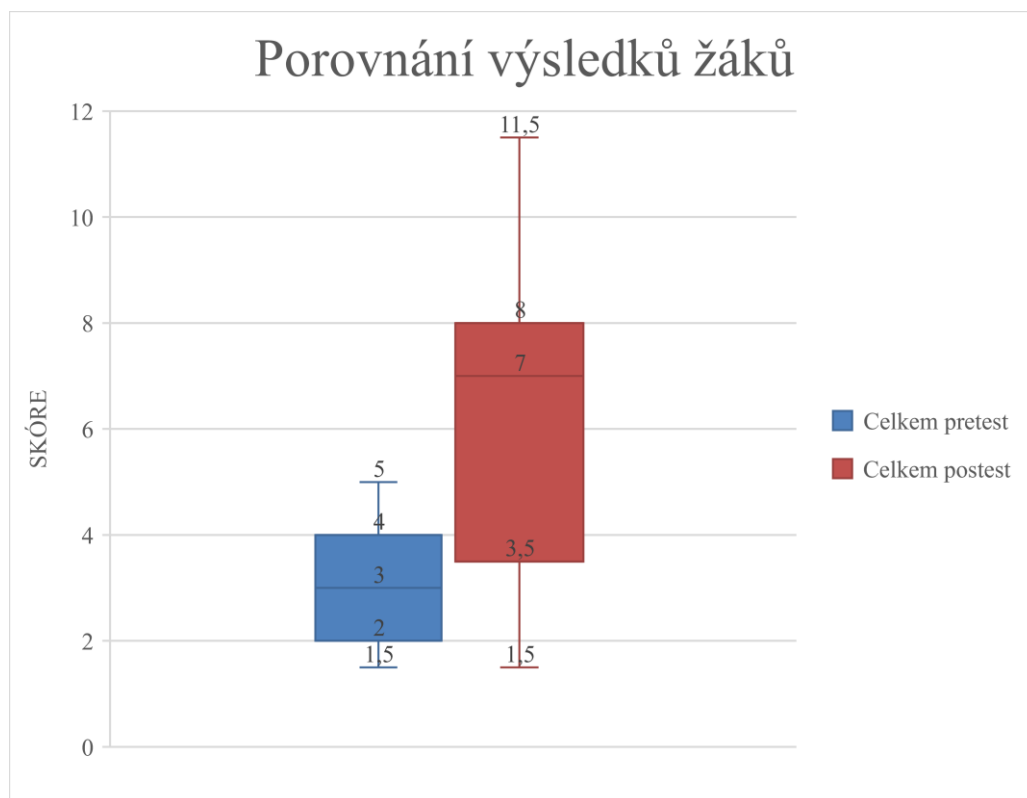
( $H_0$ ): Neexistuje významný rozdíl mezi průměrným skóre pretestu a posttestu.

Výsledek, který porovnáme s kritickou hodnotou, je T-statistika = -4,75 a dále se nám hodí p-hodnota = 0.00078. Porovnáme absolutní hodnotu T-statistiky s kritickou hodnotou a pokud je větší, tak hypotézu zamítáme.



$$|-4,75| > 2,228$$

Hodnota T-statistiky je v absolutní hodnotě větší než kritická hodnota 2,228 a velmi nízká p-hodnota = 0,00078 potvrzuje statistickou významnost tohoto zlepšení na hladině významnosti menší než 0,05. Proto tedy můžeme  $H_0$  zamítnout. To znamená, že existuje významný rozdíl mezi průměrnými skóre pretestu a post-testu. Jinými slovy, studenti se po robotickém projektu významně zlepšili.

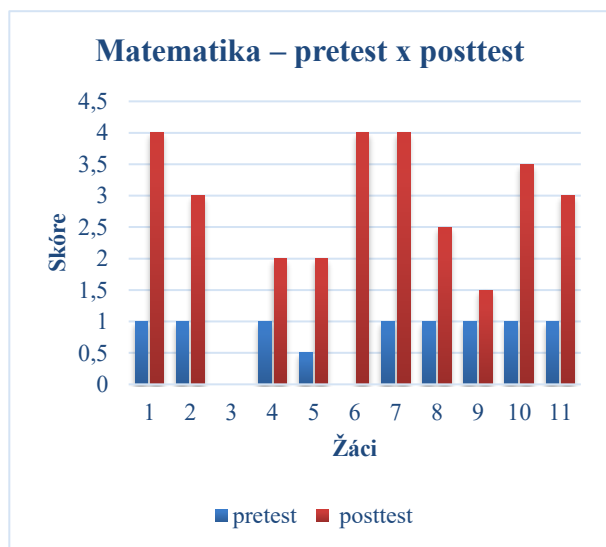


Graf 2 – Krabicový graf pretestu a posttestu

V krabicovém grafu můžeme sledovat, že medián posttest skóre je výrazně vyšší než medián pretest skóre, což naznačuje celkové zlepšení. Můžeme si povšimnout zvýšení maxima z 5 bodů na 11,5.

Dále zhlédneme a shrneme data z jednotlivých oblastí matematiky, fyziky a informatiky, kde zjistíme průměrné zlepšení, či zhoršení a co teoreticky mohlo ovlivnit tyto změny.

## Matematika

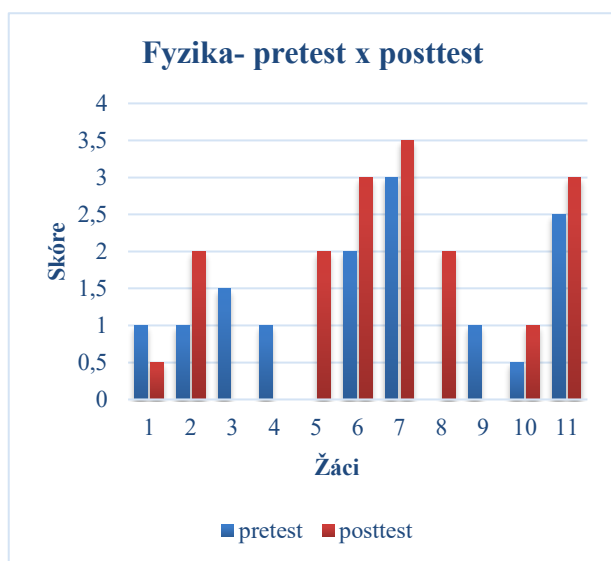


Graf 3 – Matematika pretest x posttest

Matematika – pretest x posttest			
Žáci	Pretest	Posttest	Rozdíl v testech
1	1	4	3
2	1	3	2
3	0	0	0
4	1	2	1
5	0,5	2	1,5
6	0	4	4
7	1	4	3
8	1	2,5	1,5
9	1	1,5	0,5
10	1	3,5	2,5
11	1	3	2

Tabulka 3 – Matematika pretest x posttest

## Fyzika



Graf 4 – Fyzika pretest x posttest

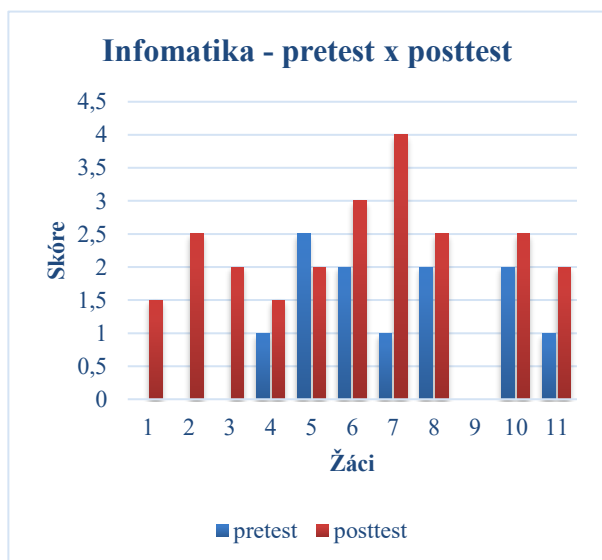
Fyzika – pretest x posttest			
Žáci	Pretest	Posttest	Rozdíl v testech
1	1	0,5	-0,5
2	1	2	1
3	1,5	0	-1,5
4	1	0	-1
5	0	2	2
6	2	3	1
7	3	3,5	0,5
8	0	2	2
9	1	0	-1
10	0,5	1	0,5
11	2,5	3	0,5

Tabulka 4 – Fyzika pretest x posttest

Z analýzy tabulky vyplývá, že účast studentů na robotickém projektu měla různé dopady na jejich výsledky fyzikálních testů. U některých studentů bylo zaznamenáno minimální zlepšení výsledků, což naznačuje, že projekt mohl mírně podpořit jejich zájem o studium fyzikálních principů. Naopak u jiných studentů byl zaznamenán pokles výsledků. Tento

pokles mohl být způsoben obtížností fyziky v rámci projektu, což vedlo ke snížení jejich motivace a následnému horšímu výkonu v testu.

## Informatika



Graf 5 – Informatika pretest x posttest

Informatika – pretest x posttest			
Žáci	Pretest	Posttest	Rozdíl v testech
1	0	1,5	1,5
2	0	2,5	2,5
3	0	2	2
4	1	1,5	0,5
5	2,5	2	-0,5
6	2	3	1
7	1	4	3
8	2	2,5	0,5
9	0	0	0
10	2	2,5	0,5
11	1	2	1

Tabulka 5 – Informatika pretest x posttest

Analýza výsledků z matematiky, fyziky a informatiky ukazuje různou míru zlepšení žáků po realizaci výuky robotického projektu. V matematice byl průměrný rozdíl mezi pretestem a posttestem 1,909 bodů, což indikuje výrazné zlepšení, kdy většina žáků vykázala pozitivní změnu. Toto zlepšení může být částečně způsobeno úlohou č. 4 s kvádrem, která spadá spíše pod informatiku jakožto 3D modelování, což mohlo výrazněji ovlivnit výstupy z informatiky. V informatice činil průměrný rozdíl 1,091 bodů, což svědčí o významném zlepšení, byť méně výrazném než v matematice. Naopak ve fyzice byl průměrný rozdíl pouze 0,318 bodů, což naznačuje, že zlepšení bylo minimální a někteří žáci se dokonce zhoršili. Tyto výsledky ukazují, že realizace robotického projektu přispěla k celkovému zlepšení žáků zejména v matematice a informatice, zatímco výsledky ve fyzice naznačují potřebu přehodnocení přístupu k výuce této oblasti.

**H<sub>3</sub>:** Realizace řešitelem navrženého robotického projektu s využitím Arduina povede k dosažení stanovených očekávaných výstupů ve STEM předmětech.

Na základě Studentova t-testu a analýzy jednotlivých oblastí přijmeme H<sub>3</sub>. Neboli realizace řešitelem navrženého robotického projektu s využitím Arduina vede k dosažení stanovených očekávaných výstupů ve STEM předmětech.

### 3.5.5 Dotazník

Dotazník zkoumá obtížnost úloh podle hodnocení žáků. Každá otázka reprezentuje jednu úlohu a jednotlivé odpovědi jsou postaveny dle Likertovy škály:

- (1) Velmi snadná: Mohl/a jsem úlohu vyřešit bez jakýchkoli problémů.
- (2) Snadná: Úloha byla jednoduchá, ale s malými obtížemi.
- (3) Střední obtížnost: Úloha měla vyváženou obtížnost, některé části byly snadné, jiné těžší.
- (4) Obtížná: Úloha byla náročná a musel/a jsem hodně přemýšlet.
- (5) Velmi obtížná: Úloha byla velmi těžká a vyžadovala hodně úsilí a přemýšlení.

Měření obtížnosti úloh								
	(1)	(2)	(3)	(4)	(5)	med	mod	sum
1. Seznámení s aplikací Tinkercad	3	5	3	-	-	2	2	22
2. Modelování základních tvarů	4	6	1	-	-	2	2	19
3. Modelování dílů pro robota	2	5	3	1	-	2	2	25
4. Určení velikosti odporů	-	1	6	4	-	3	3	36
5. Výpočet hodnoty odporu	-	2	4	2	3	3	3	39
6. Sestavení obvodu s LED diodou	1	3	6	1	-	3	3	29
7. Arduino a blikající dioda	1	4	4	2	-	3	{2, 3}	29
8. Arduino a tři LED diody	1	2	8	-	-	3	3	29
9. Princip krokového motoru	1	4	-	6	-	4	4	33
10. Funkce pro pohyb krokového motoru	1	3	3	3	1	3	{2, 3, 4}	33
11. Tisk robota	4	2	4	1	-	2	{1, 3}	24
12. Sestavení robota	5	3	3	-	-	2	1	20
13. Principy funkcí v programování	-	2	6	2	1	3	3	35
14. Vytvoření funkcí pro pohyb robota	1	4	4	1	1	3	{2, 3}	30
15. Pohyby robota	2	1	3	3	2	3	{3, 4}	35
16. Určení rychlosti robota	1	2	2	5	1	4	4	36
17. Jízda slalomem	1	1	3	3	3	4	{3, 4, 5}	39
18. Ultrazvukové čidlo	1	1	3	3	3	4	{3, 4, 5}	39
19. Měření vzdálenosti	-	2	6	1	2	3	3	36
20. Reakce na překážku	-	1	4	5	1	4	4	39

Tabulka 6 – Měření obtížnosti úloh

Tabulka obsahuje četnosti zvolených odpovědí žáků z dotazníku a tím, že se jedná o ordinální hodnoty, u kterých nedává smysl např. aritmetický průměr, jsme rozšířili tabulku o medián (střední hodnota), modus (nejčastější hodnota) a sumu, ta určuje obtížnost úlohy, jakožto součet zvolených odpovědí.

Úlohy jsou gradované, tudíž od nejsnazších po nejobtížnější: modelování – prekonceptové úlohy – programování motoru – sestavení a tisk – programování pohybu – senzory.

Na základě mediánu rozdělíme podle žakovských odpovědí úlohy do tří skupin jako snadné (2), středně obtížné (3) a obtížné (4). Tímto nám tedy odpadají okrajové hodnoty: velmi snadné a velmi obtížné úlohy.

Mezi **snadné úlohy** zařadili: seznámení s aplikací Tinkercad, modelování základních tvarů, modelování dílů pro robota, tisk robota, sestavení robota.

**Středně obtížné úlohy** představují úlohy: určení velikosti odporu, výpočet hodnoty odporu, sestavení obvodu s LED diodou, Arduino a blikající dioda, Arduino a tři LED diody, funkce pro pohyb krokového motoru, principy funkcí v programování, vytvoření funkcí pro pohyb robota, pohyby robota a měření vzdálenosti.

**Obtížné úlohy** zahrnují: princip krokového motoru, určení rychlosti robota, jízda slalomem, ultrazvukové čidlo a reakce na překážku.

Na základě provedeného hodnocení obtížnosti úloh žáky a srovnání s vlastním pozorováním z hodin, doplněného o rozhovory a vlastní vnímání obtížnosti úloh (na základě očekávaných výstupů) se objevily následující rozpory.

Úlohu „Určení velikosti odporů“ (č. 4) hodnotili žáci jako obtížnou s mediánem 3 a součtem 36 bodů. Z mého pohledu však tato úloha nebyla náročná, protože žáci při praktickém určování velikosti odporů neprojevovali výrazné potíže. Souhlasím s tím, že tato úloha může být považována za středně obtížnou, nicméně mám pochybnosti ohledně součtu 36 bodů. Tento součet naznačuje vyšší obtížnost ve srovnání s jinými úlohami, jako jsou například programovací úlohy, které mají nižší součty bodů, přestože jsou dle mého názoru výrazně náročnější.

Úloha „Výpočet hodnoty odporu“ (č. 5) byla žáky hodnocena vysokým stupněm obtížnosti s celkovým součtem 39 bodů. Podle mého názoru je tato úloha relativně jednoduchá, jelikož

vyžaduje pouze použití Ohmova zákona, tedy dosazení hodnot do vzorce. V kontextu celého projektu rozhodně nepatří mezi nejtěžší úlohy. Stejněho součtu dosáhly také úlohy týkající se sonaru, jejichž obtížnost je ve srovnání s touto úlohou nesrovnatelně vyšší.

Úlohy „Určení rychlosti robota“ (č. 16) a „Jízda slalomem“ (č. 17) byly žáky hodnoceny jako obtížné se součty 36 a 39 bodů. Podle mého názoru však byly tyto úlohy spíše nenáročné. Na základě mého pozorování jsem měl dojem, že tyto úlohy žáky bavily a zvládli je bez větších problémů. Jednalo se především o praktické měření a využití jednoduchého vzorce pro dráhu. U úlohy „Jízda slalomem“ je možné, že žáci nesprávně předpokládali, co vše je nutné vytvořit pro splnění této úlohy, jako například všechny funkce pro pohyb robota. Tyto funkce však byly již vytvořeny v předchozích úlohách.

Shodli jsme se obecně v úlohách zaměřených na modelování: „modelování základních tvarů“ (č. 2) a „modelování dílů pro robota“ (č. 3) byly hodnoceny jako snadné, což koresponduje s mými zkušenostmi. Stavba modelů byla pro žáky spíše snadná, což odpovídá i hodnocení, kde medián a součet hodnot odpovídají mé vypořizované úrovni obtížnosti. Programovací úlohy byly žáky hodnoceny jako středně obtížné až obtížné. Zde se ukazuje, že programování bylo pro žáky největším problémem, což je v souladu s mým pozorováním, že právě programovací úkoly představovaly výrazně vyšší náročnost než modelování a stavba robota.

Na základě rozhovorů jsem zjistil, že žáci často nemohou přesně rozhodnout, které úlohy byly nejobtížnější. Jednou z možností je, že si již přesně nepamatovali všechny detaily úkolů, nebo neměli dostatečné pochopení a představu o jednotlivých úlohách. Je také třeba zdůraznit, že určování obtížnosti úloh není jednoduchý proces.

**H<sub>2</sub>:** Žáci budou schopni identifikovat konkrétní problémy a obtíže, které vyvstanou při řešení interdisciplinárních úloh v rámci řešitelem navrženého robotického projektu.

Závěrem k ověření hypotézy H<sub>2</sub> musíme vycházet z dat získaných z pozorování, rozhovorů a dotazníku. Z pozorování lze sledovat míru samostatnosti žáků při řešení úloh. Některé úlohy žáci nebyli schopni samostatně identifikovat a vyřešit, a museli být instruováni. Existovaly však i skupiny, které dokázaly úlohy vyřešit samostatně a identifikovat problémy bez pomoci. Obecně však nelze tuto samostatnost zobecnit na všechny žáky. Podobně

z rozhovorů bylo patrné, že žáci často nedokázali určit, které úlohy byly nejobtížnější, a většina se spíše zaměřila na obecné problémy s programováním. Dotazník nám ukazuje, že žáci nemají jasnou představu o obtížnosti jednotlivých úloh.

Na základě těchto zjištění tedy druhou hypotézu  $H_2$  zamítáme. Žáci tedy nejsou schopni identifikovat konkrétní problémy a obtíže, které vyvstanou při řešení interdisciplinárních úloh v rámci řešitelem navrženého robotického projektu.

# Závěr

Cílem práce bylo v teoretické části vymežit základní pojmy související s robotikou, konceptem STEM, mikrokontrolery Arduino a konstruktivistické metody pro výuku edukační robotiky. Empirická část zahrnovala vytvoření námětů úloh, které byly ověřeny na základní škole prostřednictvím případové studie. Na základě této studie bylo možné odpovědět na výzkumné otázky a vyhodnotit jednotlivé hypotézy. Hlavní výzkumná otázka zněla: jaká je úspěšnost a efektivita žáků při řešení navržené sady úloh a zda jsou žáci schopni identifikovat a srozumitelně formulovat konkrétní problémy spojené s řešením těchto úkolů. Ve vztahu k této otázce byly formulovány tři hypotézy, dvě z nich byly přijaty ( $H_1$ ,  $H_3$ ) a jedna z nich zamítnuta ( $H_2$ ).

$H_1$ : Žáci 9. ročníku ZŠ budou schopni sestavit a naprogramovat robotický systém s využitím Arduina a disponují dostatečnými znalostmi a dovednostmi řešit typ úkolů, které jsou součástí řešitelem připravené sady úloh.

$H_2$ : Žáci budou schopni identifikovat konkrétní problémy a obtíže, které vyvstanou při řešení interdisciplinárních úloh v rámci řešitelem navrženého robotického projektu.

$H_3$ : Realizace řešitelem navrženého robotického projektu s využitím Arduina povede k dosažení stanovených očekávaných výstupů ve STEM předmětech.

Z analýzy dat případové studie byly ověřeny jednotlivé hypotézy. Potvrdilo se, že žáci jsou schopni vytvořit, sestavit a naprogramovat robotický systém využívající Arduino, že disponují dostatečnými znalostmi a dovednostmi k řešení úkolů obsažených v připravené sadě úloh. Dále bylo potvrzeno, že realizace robotického projektu se žáky vede k dosažení stanovených očekávaných výstupů v předmětech STEM. Naopak se ukázalo, že žáci nejsou ve všech případech schopni identifikovat problémy a své obtíže, které vyvstávají při řešení úloh.

Na základě hypotéz odpovíme na výzkumnou otázku. Výzkum odhalil, že žáci mají obtíže s identifikací a formulací vlastních problémů při řešení úloh. To jsme ověřili na základě dotazníku, kde žáci hodnotili obtížnost úloh, dále na základě polostrukturovaných rozhovorů a pozorování. Jejich odpovědi z dotazníku sledující obtížnost úlohy nekorespondovaly



s předpokladem námi vytvořených gradovaných úloh a jejich obtížností. Druhým zdrojem dat pro toto konstatování byly rozhovory s žáky, kde jsme pozorovali, že nedokážou přesně zodpovědět, jaká oblast nebo úloha je pro ně problematická. Pokud bychom se měli vyjádřit k efektivitě a úspěšnosti žákovských řešení úloh, tak zásadním výstupem, který může poukazovat na značnou efektivitu, je, že žáci zvládli celý projekt za půl roku. Samozřejmě pokud by byla časová dotace vyšší, přibližně jeden školní rok, máme za to, že by pro žáky a kvalitu výuky byla přínosná. Úspěšnost žáků lze posoudit na základě jejich samostatnosti při řešení úloh, přičemž průměrná míra samostatného řešení úloh dosáhla 63 %. Tuto hodnotu můžeme dokladovat na základě pozorování z hodin. To můžeme považovat za úspěch, jelikož jsme předpokládali, že žáci nevykážou nadprůměrnou míru samostatnosti.

Musíme však upozornit na situace, kdy žáci nebyli schopni pokračovat bez instrukcí učitele. Jednalo se primárně o oblast programování jednotlivých funkcí robota. Hlavním problémem byla právě syntax programovacího jazyka, jelikož žáci neměli žádné zkušenosti s programováním. Na základě rozhovoru s žáky a pozorování je varianta blokového programování schůdnější variantou, než textový zápis algoritmu.

Z analýzy pretestů a posttestů můžeme sledovat průměrně dvojnásobné bodové zlepšení. Žáci se zlepšili ve všech oblastech, nejvíce v matematice a informatice. Ve fyzice průměrně o pouhých 0,3 bodu. Někteří žáci rovnou úlohy z fyziky v posttestu přeskočili. Tento jev může být ovlivněn obtížností fyziky, která mohla mít na žáky negativní vliv a snížila jejich sebevědomí v této oblasti. Na základě výsledků z testu, můžeme konstatovat, že realizace projektu vedla k minimálně průměrnému, či vyššímu osvojení očekávaných výstupů a ke zlepšení žákovských znalostí v jednotlivých předmětech.

Na základě uskutečněného projektu se nabízí do budoucna možnost různých navazujících a rozšiřujících aktivit. Návrh, konstrukce, stavba a programování robotických vozidel může mít do budoucna přesah pro výuku jednotlivých předmětů. Z počátku by se mohly dokončit a do hloubky probrat úlohy, které dvě skupiny stihly částečně a tři s nimi ani nezačaly. Jednalo se o úlohy s fotorezistory a senzory pro měření vzdálenosti. Další praktickou aplikací mohou být robotická vozidla jako situační modely při výuce fyziky a matematiky. Například při řešení úloh zaměřených na pohyb mohou žáci použít roboty k vizualizaci a ověření výsledků, přičemž úlohy by měly obsahovat zadané rychlosti robotů,

aby odpovídaly probíraným situacím. Další aplikací robotů k výuce fyziky může být měření veličin, které jsou obtížně dostupné nebo nebezpečné pro člověka, například měření teploty ohně pomocí robotického modelu hasičského auta, který by mohlo simulovat hašení „požáru“ na základě naměřené teploty.

## Seznam použitých informačních zdrojů

ARDUINO – příručka programátora. Online. DPS. 2021. Dostupné z: <https://www.dps-az.cz/getFile/id:84163/Arduino-p%C5%99%C3%ADru%C4%8Dka-program%C3%A1tora.pdf>. [cit. 2024-05-03].

ASIMOV, Isaac. *I, robot*. Robot series. New York: Bantam Books, 2008. ISBN 978-0-553-38256-3.

ASIMOV, Isaac. *Robots and Empire*. Argo, 2012. ISBN 978-80-257-0720-3.

BLUM, Jeremy. *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. 2. John Wiley, 2020. ISBN 978-1-119-40535-1.

*Cinema 4D Release 6*. Computer Press, 2004. ISBN 80-251-0001-4.

CO JE TO LED DIODA? Online. LEDme. 2024. Dostupné z: <https://ledme.cz/textove-novinky/clanky/co-je-LED-dioda>. [cit. 2024-05-02].

Co je to mikrokontrolér? Online. Zonepi. 2019. Dostupné z: <https://blog.zonepi.cz/maker-uno-ve-vyuce-1-5-co-je-to-mikrokontroler/>. [cit. 2024-05-02].

ČAPEK, Karel. *R.U.R.: Rossum's universal robots*. V nakladatelství Artur vydání čtvrté. D (Artur). Praha: Artur, 2021. ISBN 978-80-7483-150-8.

*Elektromotory – Synchronní vládné*. Online. Automobil. 2022. Dostupné z: [https://www.automobilrevue.cz/rubriky/presunuto-na-trucker-cz-truck-bus/predstavujeme/elektromotory-synchronni-vladne\\_48530.html](https://www.automobilrevue.cz/rubriky/presunuto-na-trucker-cz-truck-bus/predstavujeme/elektromotory-synchronni-vladne_48530.html). [cit. 2024-04-28].

EWING-CHOW, Daphne. *Here Are Five Global Restaurants Staffed By Robot Chefs*. Online. Forbes. 2024. Dostupné z: <https://www.forbes.com/sites/daphneewingchow/2024/03/31/here-are-five-global-restaurants-staffed-by-robot-chefs/>. [cit. 2024-04-30].

HEMMENDINGER, David. *Logo*. Online. Britannica. 2024. Dostupné z: <https://www.britannica.com/technology/Logo>. [cit. 2024-04-30].

HLAVÁČ, Václav. *SENZORY PRO ROBOTIKU*. Online. ČVUT. 2024. Dostupné z: <https://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/51SenzoryRobotika.pdf>. [cit. 2024-05-02].

HLAVÁČ, Václav. *Úvod do robotiky*. Online. 2024. Dostupné z: <https://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/01UvodRobotika.pdf>. [cit. 2024-04-30].

*Jak funguje krokový motor?* Online. Raveo. 2024. Dostupné z: <https://www.raveo.cz/aktualita/jak-funguje-krokovy-motor/>. [cit. 2024-04-28].

*Klíčové termíny v 3D tisku a jejich význam*. Online. 3D Stisk. 2024. Dostupné z: <https://3dstisk.cz/klicove-terminy-v-3d-tisku-a-jejich-vyznam/>. [cit. 2024-06-19].

*Koncepce dovedností STEM má přispět ke zvýšení mezinárodní konkurenceschopnosti Austrálie*. Online. NPI. 2019. Dostupné z: <https://archiv-nuv.npi.cz/vystupy/koncepce-dovednosti-stem-ma-prispet-ke-zvyseni-mezinarodni.html>. [cit. 2024-03-10].

*Koncept STEM*. Online. NPI. 2019. Dostupné z: <https://archiv-nuv.npi.cz/p-kap/koncept-stem.html>. [cit. 2024-01-10].

*Kooperativní výuka*. Online. RVP. 2011. Dostupné z: [https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD\\_lexikon/K/Kooperativn%C3%AD\\_v%C3%BDuka](https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD_lexikon/K/Kooperativn%C3%AD_v%C3%BDuka). [cit. 2024-06-10].

KOŠÁTKOVÁ, Markéta. *Konstruktivismus*. Online. MUNI. 2023. Dostupné z: [https://is.muni.cz/do/rect/el/estud/pedf/ps23/pvc/web/pages/03\\_01\\_05\\_konstruktivismus.html](https://is.muni.cz/do/rect/el/estud/pedf/ps23/pvc/web/pages/03_01_05_konstruktivismus.html). [cit. 2024-04-29].

KUPILÍKOVÁ, Martina. *Arduino na ZŠ*. Online. Itfitness. 2020. Dostupné z: <http://itfitness.cz/arduino-na-zs/>. [cit. 2024-06-11].

KUŘINA, František. *10 geometrických transformací*. PROMETHEUS, 2010. ISBN 80-7196-231-7.

*Kvalitativní a kvantitativní výzkum, vzájemné porovnání*. Online. Wikisofia. 2024. Dostupné z: [https://wikisofia.cz/wiki/3.\\_Kvalitativn%C3%AD\\_a\\_kvantitativn%C3%AD\\_v%C3%BDzkum,\\_vz%C3%A1jemn%C3%A9\\_porovn%C3%A1n%C3%AD](https://wikisofia.cz/wiki/3._Kvalitativn%C3%AD_a_kvantitativn%C3%AD_v%C3%BDzkum,_vz%C3%A1jemn%C3%A9_porovn%C3%A1n%C3%AD). [cit. 2024-06-21].

LU, Chow-Chin. *Elementary School Students Learn Arduino Programming to Assemble Sensory-Controlled Works*. Online. 2020. Dostupné z: <https://www.ijiet.org/vol110/1374-IT006.pdf>. [cit. 2024-06-11].

MATIČKA, Robert. *KONSTRUKCE PRŮMYSLOVÝCH ROBOTŮ A MANIPULÁTORŮ*. ČVUT, 1995. ISBN 80-01-01291-3.

MISRA, Yogesh. *Programming and Interfacing with Arduino*. CRC PRESS, 2021. ISBN 978-1032063164.

NOVÁK, Milan a PECH, Jiří. *Robotika pro střední školy: programujeme Arduino*. Jihočeská univerzita v Českých Budějovicích, 2018. ISBN 978-80-7394-786-6.

*Odvození transformačních matic pro různé typy rotace*. Online. MUNI. 2013. Dostupné z: [https://mineralogie.sci.muni.cz/kap\\_1\\_3\\_symetrie/rotace\\_priklad.htm](https://mineralogie.sci.muni.cz/kap_1_3_symetrie/rotace_priklad.htm). [cit. 2024-04-02].

PAUL, Richard. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Cambridge, MA: MIT Press, 1981

PIAGET, Jean. *The Principles of Genetic Epistemology*. Psychology Press, 1997. ISBN 978-0710072962.

*Polygonal Mesh*. Online. ScienceDirect. 2018. Dostupné z: <https://www.sciencedirect.com/topics/computer-science/polygonal-mesh>. [cit. 2024-04-02].

ŘEHÁČEK, Jan. *Jak zatočit s virtuálním světem*. Online. Matfyz. 2019. Dostupné z: <https://www.matfyz.cz/clanky/matykani-xxvii-jak-zatocit-s-virtualnim-svetem>. [cit. 2024-04-02].

ŘEZÁČ, Kamil. *Krokové motory*. Online. Robotika. 2002. Dostupné z: <https://robotika.cz/articles/steppers/cs>. [cit. 2024-03-27].

*Science, technology, engineering, and mathematics*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024. Dostupné z: [https://en.wikipedia.org/wiki/Science,\\_technology,\\_engineering,\\_and\\_mathematics](https://en.wikipedia.org/wiki/Science,_technology,_engineering,_and_mathematics). [cit. 2024-03-20].

- SERAFÍN, Čestmír. *CELOŽIVOTNÍ UČENÍ A KONSTRUKTIVISMUS*. Online. LifeLongLearningMendelu. 2011. Dostupné z: [https://lifelonglearning.mendelu.cz/media/pdf/LLL\\_20110102007.pdf](https://lifelonglearning.mendelu.cz/media/pdf/LLL_20110102007.pdf). [cit. 2024-04-29].
- STEM/MINT koncept vzdělávání 2023*. Online. Insgraf. 2023. Dostupné z: <https://insgraf.cz/katalog/katalog-STEAM-MINT-koncept-vzdelavani-2023/II/index.html>. [cit. 2024-01-10].
- ŠŤASTNÝ, František. *Senzory*. Online. Amper MUNI. 1997. Dostupné z: [https://amper.ped.muni.cz/jenik/nejistoty/html\\_tree/node16.html](https://amper.ped.muni.cz/jenik/nejistoty/html_tree/node16.html). [cit. 2024-05-02].
- TOCHÁČEK, Daniel a LAPeŠ, Jakub. *Edukační robotika*. Praha: Pedagogická fakulta Univerzity Karlovy, 2012. ISBN 978-80-7290-577-5.
- URBANOVÁ, Libuše. *Umělá inteligence jako součást robotiky a její využití v dnešním světě*. Online. Alfun. 2023. Dostupné z: <https://www.alfun.cz/umela-inteligence-jako-soucast-robotiky-a-jeji-vyuziti-v-dnesnim-svete/>. [cit. 2024-04-29].
- VOJTA, Václav. *Arduino v 6. třídě*. Online. Wix. 2024. Dostupné z: <https://vkp032.wixsite.com/arduino/fyzika>. [cit. 2024-06-11].
- Výukové metody aktivizující*. Online. RVP. 2012. Dostupné z: <https://clanky.rvp.cz/clanek/c/s/15017/VYUKOVE-METODY-AKTIVIZUJICI.html>. [cit. 2024-06-10].
- Vztah vývoje a učení, školní vyučování, zóna nejbližšího vývoje*. Online. Portál. 2004. Dostupné z: <https://nakladatelstvi.portal.cz/nakladatelstvi/aktuality/78984>. [cit. 2024-06-11].
- What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016, s. 3–5. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/_data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].
- ŽÁRA, Jiří; BENEŠ, Bedřich; SOCHOR, Jiří; FELKEL, Petr. *Moderní počítačová grafika*. Computer Press, 2005. ISBN 80-251-0454-0.

## Obrázky

ARDUINO UNO R3 – Upgraded Learning Kit. Online. PS electronic. 2024. Dostupné z: <https://www.pselectronic.cz/index.php?action=k5505-155-arduino-uno-r3-upgraded-learning-kit>. [cit. 2024-05-02].

EWING-CHOW, Daphne. *Here Are Five Global Restaurants Staffed By Robot Chefs*. Online. Forbes. 2024. Dostupné z: <https://www.forbes.com/sites/daphneewingchow/2024/03/31/here-are-five-global-restaurants-staffed-by-robot-chefs/>. [cit. 2024-04-30].

KROKOVÝ MOTOR – CO TO JE A JAK FUNGUJE. Online. EUfactory. 2018. Dostupné z: [https://shop.eufactory.com/blog/12\\_krokovy-motor-uvod](https://shop.eufactory.com/blog/12_krokovy-motor-uvod). [cit. 2024-04-28].

LU, Chow-Chin. *Elementary School Students Learn Arduino Programming to Assemble Sensory-Controlled Works*. Online. 2020. Dostupné z: <https://www.ijiet.org/vol10/1374-IT006.pdf>. [cit. 2024-06-11].

PISA 2022 Worldwide Ranking. Online. FactsMaps. 2023. Dostupné z: <https://factsmaps.com/pisa-2022-worldwide-ranking-average-score-of-mathematics-science-and-reading/>. [cit. 2024-03-20].

RIMMOND, Scott. *What is it like to be a turtle?* Online. Medium. 2017. Dostupné z: <https://medium.com/thinking-with-computers/what-is-it-like-to-be-a-turtle-829eabb129>. [cit. 2024-04-30].

ŘEZÁČ, Kamil. *Krokové motory*. Online. Robotika. 2002. Dostupné z: <https://robotika.cz/articles/steppers/cs>. [cit. 2024-03-27].

STEM/MINT koncept vzdělávání 2023. Online. Insgraf. 2023. Dostupné z: <https://insgraf.cz/katalog/katalog-STEAM-MINT-koncept-vzdelavani-2023/II/index.html>. [cit. 2024-01-10].

What is STEM? The need for unpacking its definitions and applications. Online. In: Adelaide SA 5000: NCVER, 2016, s. 3–7. ISBN 978 1 925173 66 6. Dostupné z: [https://www.ncver.edu.au/\\_data/assets/pdf\\_file/0023/61349/What-is-STEM.pdf](https://www.ncver.edu.au/_data/assets/pdf_file/0023/61349/What-is-STEM.pdf). [cit. 2024-03-18].

## Seznam obrázků

Obr. 1 – Rozložení potenciálně nahraditelných pozic roboty v restauracích .....	12
Obr. 2 – Konstrukce čtverce pomocí želví grafiky.....	18
Obr. 3 – Návrh STEM učebny.....	22
Obr. 4 – PISA hodnocení 2022.....	23
Obr. 5 – The House of STEM.....	25
Obr. 6 – Schéma STEM.....	26
Obr. 7 – Booleovské operace ve 3D grafice.....	29
Obr. 8 – HPB souřadnice.....	33
Obr. 9 – Popis Arduino desky .....	35
Obr. 10 – Popis STEM projektu .....	37
Obr. 11 – Elektronické součástky pro sadu Arduino UNO .....	38
Obr. 12 – Krokový motor .....	40
Obr. 13 – Dělení krokových motorů.....	41
Obr. 14 – Proces jedné otočky krokového motoru .....	42
Obr. 15 – Rozklad Arduino jazyku.....	43
Obr. 16 – Robotické Arduino Auto .....	48
Obr. 17 – 3D model podvozku Arduino auta .....	51
Obr. 18 – 3D model kol a stříšky Arduino auta .....	52
Obr. 19 – PrusaSlicer.....	53
Obr. 20 – Schéma zapojení Arduino a el. součástek robota .....	54
Obr. 21 – Schéma pro určení hodnoty rezistoru.....	61
Obr. 22 – Schéma zapojení LED diody a rezistoru .....	62
Obr. 23 – Žákovské výtvary robotických aut.....	75
Obr. 24 – Test, 3D modelování .....	85
Obr. 25 – Test – algoritmické myšlení .....	86



## Seznam tabulek

Tabulka 1 – Výsledky z úloh z pohledu samostatnosti při řešení .....	76
Tabulka 2 – Celkové výsledky z pretestu a posttestu.....	87
Tabulka 3 – Matematika pretest x posttest .....	89
Tabulka 4 – Fyzika pretest x posttest .....	89
Tabulka 5 – Informatika pretest x posttest .....	90
Tabulka 6 – Měření obtížnosti úloh.....	91

## Seznam grafů

Graf 1 – Žákovská úspěšnost řešení úloh .....	77
Graf 2 – Krabicový graf pretestu a posttestu.....	88
Graf 3 – Matematika pretest x posttest.....	89
Graf 4 – Fyzika pretest x posttest .....	89
Graf 5 – Informatika pretest x posttest .....	90

# Přílohy

## Žákovský kód

```
int L1=2; // motor 1 je na pinu 2
int L2=3;
int L3=4;
int L4=5;

int P1=6; // to stejny akorat pravej ok?
int P2=7;
int P3=8;
int P4=9;

int i=0;
int ujet_mm=0;
long vzdalenost;
int vysilac;
int prijimac;

void setup() {
  // put your setup code here, to run once:
  pinMode(L1,OUTPUT);
  pinMode(L2,OUTPUT);
  pinMode(L3,OUTPUT);
  pinMode(L4,OUTPUT);
  pinMode(P1,OUTPUT);
  pinMode(P2,OUTPUT);
  pinMode(P3,OUTPUT);
  pinMode(P4,OUTPUT);
}

void jizda_vzad() {
  ;digitalWrite(L1,HIGH); digitalWrite(L2,LOW);digitalWrite(L3,LOW);digitalWrite(L4,LOW);
  delay(1)
  ;digitalWrite(L1,HIGH);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW);
  delay(1)
  ;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW);
  delay(1)
  ;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
  delay(1)
  ;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
  delay(1)
  ;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,HIGH);
  delay(1)
  ;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
  delay(1)
  ;digitalWrite(L1,HIGH);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
  delay(1)

  ;digitalWrite(P1,HIGH); digitalWrite(P2,LOW);digitalWrite(P3,LOW);digitalWrite(P4,LOW);
```

```

delay(1)
;digitalWrite(P1,HIGH);digitalWrite(P2,HIGH); digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH); digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH); digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,HIGH);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1);
;digitalWrite(P1,HIGH);digitalWrite(P2,LOW); digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1);
}

void otocka_vpred() {
;digitalWrite(L1,HIGH); digitalWrite(L2,LOW);digitalWrite(L3,LOW);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,HIGH);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW) ;
delay(1) ;
digitalWrite(L1,HIGH);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW) ;
delay(1)

;digitalWrite(P1,HIGH); digitalWrite(P2,LOW);digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,HIGH);digitalWrite(P2,LOW); digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,HIGH);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH); digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH); digitalWrite(P3,LOW);digitalWrite(P4,LOW) ;
delay(1) ;
digitalWrite(P1,HIGH);digitalWrite(P2,HIGH); digitalWrite(P3,LOW);digitalWrite(P4,LOW) ;
delay(1);
}

```

```

void otocka_vlevo (){
;digitalWrite(L1,HIGH);  digitalWrite(L2,LOW);digitalWrite(L3,LOW);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,HIGH);digitalWrite(L2,LOW);  digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW);  digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW);  digitalWrite(L3,HIGH);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW);  digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH);  digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH);  digitalWrite(L3,LOW);digitalWrite(L4,LOW) ;
delay(1) ;
digitalWrite(L1,HIGH);digitalWrite(L2,HIGH);  digitalWrite(L3,LOW);digitalWrite(L4,LOW) ;
delay(1)

;digitalWrite(P1,HIGH);  digitalWrite(P2,LOW);digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,HIGH);digitalWrite(P2,HIGH);  digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH);  digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH);  digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,HIGH);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1);
;digitalWrite(P1,HIGH);digitalWrite(P2,LOW);  digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1);
}

void otocka_vpravo(){
    digitalWrite(P1,HIGH);  digitalWrite(P2,LOW);digitalWrite(P3,LOW);digitalWrite(P4,LOW
);
delay(1)
;digitalWrite(P1,HIGH);digitalWrite(P2,LOW);  digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,LOW);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,HIGH);digitalWrite(P4,HIGH);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW);  digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH);  digitalWrite(P3,HIGH);digitalWrite(P4,LOW);
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,HIGH);  digitalWrite(P3,LOW);digitalWrite(P4,LOW) ;
delay(1) ;
digitalWrite(P1,HIGH);digitalWrite(P2,HIGH);  digitalWrite(P3,LOW);digitalWrite(P4,LOW) ;

```

```

delay(1)

;digitalWrite(L1,HIGH); digitalWrite(L2,LOW);digitalWrite(L3,LOW);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,HIGH);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,LOW);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,HIGH); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,LOW);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,HIGH);digitalWrite(L4,HIGH);
delay(1)
;digitalWrite(L1,LOW);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1);
;digitalWrite(L1,HIGH);digitalWrite(L2,LOW); digitalWrite(L3,LOW);digitalWrite(L4,HIGH);
delay(1);
}
void vypni_motory(){
    digitalWrite(L1,LOW); digitalWrite(L2,LOW);digitalWrite(L3,LOW);digitalWrite(L4,LOW)
;
delay(1)
;digitalWrite(P1,LOW);digitalWrite(P2,LOW); digitalWrite(P3,LOW);digitalWrite(P4,LOW);
delay(1);
}

void ujet_vzdalenost(int mm) {
    int otacek;
    int j=0;
    otacek=(int)round(mm*1.46);
for(j=0;j<otacek;j++) {
    otopka_vpřed();

}

}

void otopka_o_uhel_vlevo(int uhel) {
    int otacek;
    int j=0;
    otacek=(int)round(uhel*2.06);
for(j=0;j<otacek;j++) {
    otopka_vlevo();

}

}

void otopka_o_uhel_vpravo(int uhel) {
    int otacek;

```

```

    int j=0;
    otacek=(int)round(uhel*2.06);
    for(j=0;j<otacek;j++) {
        otopka_vpravo();
    }
}

void Zmer_Vzdalenost(){ // určuje vzdálenost v mm
    long mereni;
    digitalWrite(vysilac, LOW);
    delayMicroseconds(20);
    digitalWrite(vysilac, HIGH);
    delayMicroseconds(30);
    digitalWrite(vysilac, LOW);
    delayMicroseconds(20);
    mereni=pulseIn(prijimac,HIGH,2000);
    vzdalenost=mereni*17/100;
}

void loop() {
    Zmer_Vzdalenost();
    if (vzdalenost>200) {otopka_vpřed();}
    else { otopka_o_uhel_vpravo(120);}
}

```

# Žákovské testy

## Posttest

### Matematika

$$\begin{array}{r} 4,78 \\ 45 \\ \hline 215,1 \end{array}$$

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?


$25^2 \cdot \pi = 157,1$        $700 / 157,1 = 4,455$  *KOLO*
2. Pokud se motor robotického auta otočí 512krát, celý robot změní svůj směr o 107°. Kolikrát se motor musí otočit, aby se robot otočil o 45°?

$512 / 107 = 4,78 \cdot 45 = 215,133$
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

$250 / 36,2 = 6,9$        $512 / 36,2 = 14,14$
4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?

$20 \cdot 30 = 6000 \cdot 10 = 60000$

$16 \text{ A } 26$



### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je 10kΩ a vstupní napětí je 5 V?

$241 \mu A$
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?

$10 k\Omega$
7. Jak poznáš, jak velký má rezistor odpor?

*PODLE BARVY*
8. Vysvětli fungování fotorezistoru.

*VŮŠÍ VÍŠIČNÍ ODKAPU*
9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátí za 4 s. Počítej s rychlostí zvuku 343 m/s.

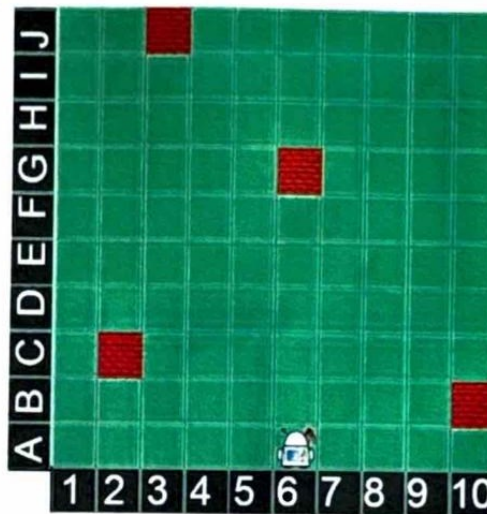
$2 \cdot 343 = 686$

68

1972

## Programování

10. Vysvětli pojem algoritmus. *NECO COSE OPAKUJE*
11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zed), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.
- 0,5 ✓ a) Dostane se někdy robot na pole C3? *NE*
- x b) Na jakém poli robot skončí? *7J*



12. Vysvětli, k čemu se využívá příkaz if.

x *IF - JE PŘÍKAZ ABY NECO UPRADL*

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

✓ *CYKLU ABY OPAKOVAL ZADANÝ VÝKOD*



## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?  
 $700 : 25 = 28$
2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o 107°. Kolikrát se motor musí otočit, aby se robot otočil o 45°?  
 $512 \cdot 107 = 54784$   
 $54784 : 45 = 1217,44$   
 $\approx 1217$
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.  
 $250 \cdot 36,2 = 9100$   
 $9100 : 512 = 17,77$   
 $\approx 18$
4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kváдру vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



$20 \cdot 30 = 6000$   
 šířka 18 cm  
 délka 26 cm

### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je 10kΩ a vstupní napětí je 5 V?  
 $I = \frac{U}{R} = \frac{5}{10000} = 0,0005 \text{ A} = 0,5 \text{ mA}$
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?  
 $R = \frac{U}{I} = \frac{5}{0,002} = 2500 \Omega = 2,5 \text{ k}\Omega$
7. Jak poznáš, jak velký má rezistor odpor?
8. Vysvětli fungování fotorezistoru. Vzniká tím více voltů elektroni  
 když je dopadla více světla
9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátí za 4 s. Počítej s rychlostí zvuku 343 m/s.

$343 \cdot 4 = 1372$   
 $1372 : 2 = 686$   
 $686 \text{ cm}$

7,56

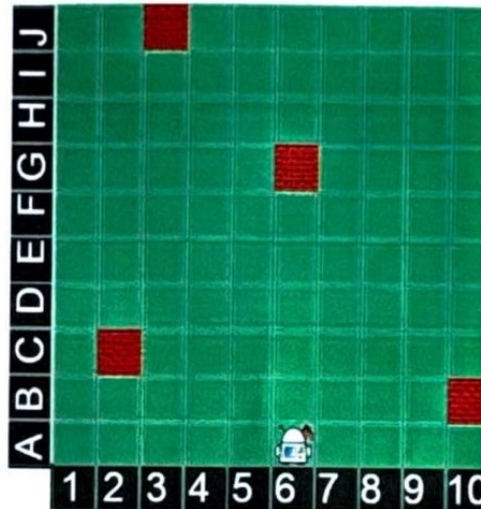
## Programování

✓ 10. Vysvětli pojem algoritmus.  
*představa / postup*

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

✓ a) Dostane se někdy robot na pole C3? *Ne*

✗ b) Na jakém poli robot skončí? *F10*



✗ 12. Vysvětli, k čemu se využívá příkaz if.  
*aby se modifikovaly příkazy*

0,5 13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.  
*loop*

## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?  
X
2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?  
X
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.
4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10\text{k}\Omega$  a vstupní napětí je 5 V?  
X  
 $V = R \cdot I$
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?  
X
7. Jak poznáš, jak velký má rezistor odpor?  
X
8. Vysvětlí fungování fotorezistoru.  
X
9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.  
X

28

## Programování

10. Vysvětli pojem algoritmus.

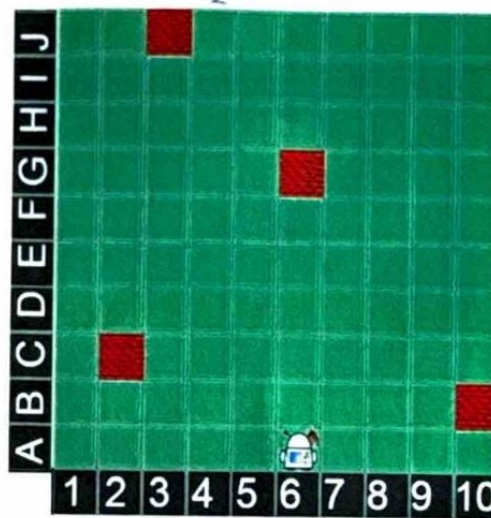
✓

~ návod kterým lze vyřešit typ úlohy

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

X a) Dostane se někdy robot na pole C3? ANO

X b) Na jakém poli robot skončí?



12. Vysvětli, k čemu se využívá příkaz if.

0,5

~ zapisování podmínek pomocí klíčového slova IF.

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

✓

~ Python ~~podobně~~ ~ opakování příkazů slouží zavádění cyklů FOR a CYKLU WHILE



## Posttest

$$\frac{157}{5} = 31.4$$

$$\frac{157}{4} = 39.25$$

### Matematika

$$7m = 700cm$$

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?

$$\frac{6,28}{25} = 0,251$$

$$\frac{1570}{0,251} = 6215$$



$$C = 2\pi r$$

$$C = 6,28 \cdot 25$$

$$C = 157cm$$

$$\frac{700}{157} = 4,4$$

$$4,4 \times$$

2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?

3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kváдру vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



$$10cm - 2cm$$

$$\text{šířka } 16cm$$

$$\text{délka } 26cm$$

### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10k\Omega$  a vstupní napětí je 5 V?
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?
7. Jak poznáš, jak velký má rezistor odpor?
8. Vysvětli fungování fotorezistoru.
9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

$$3,15 A$$

## Programování

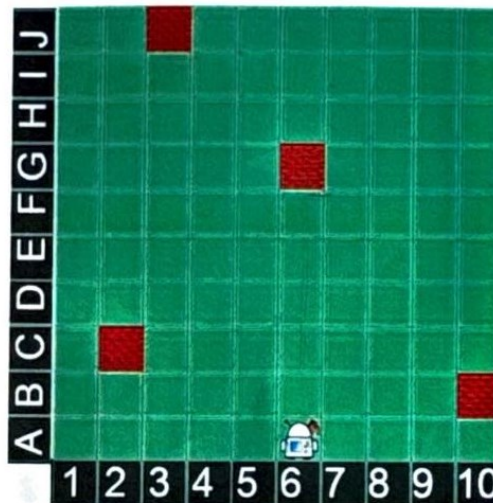
10. Vysvětli pojem algoritmus.

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

✓ a) Dostane se někdy robot na pole C3? *ne*

✗ b) Na jakém poli robot skončí?

*0,5*



12. Vysvětli, k čemu se využívá příkaz if.

*0,5* znamená to jestliže nebo když

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

*0,5* void/loop/

## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?  
X 14 krát
2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?  
X
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.  
✓1 = 3500 KRÁT 14 KRÁT
4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádru vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?  
✓1 16 a 26 VÝŠKA 8cm



### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10\text{ k}\Omega$  a vstupní napětí je 5 V?  
X 10  $\mu$ A
6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?  
X
7. Jak poznáš, jak velký má rezistor odpor?  
✓1 PODLE VZORCE  $R=U/I$
8. Vysvětli fungování fotorezistoru.  
✓1 SVĚTLO NARAZÍ A PŘEDÁ SVOJI ENERGIÍ
9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

68

## Programování

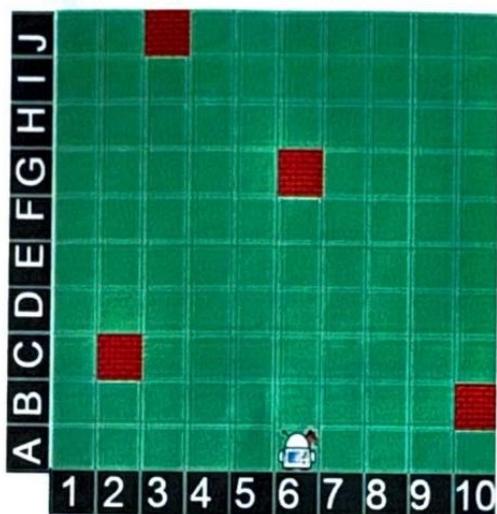
✓ 1

10. Vysvětli pojem algoritmus. PŘESNÝ NÁVOD POSTUPU, KTERÝM LZE NĚCO VYŘEŠIT

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zed), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

X

a) Dostane se někdy robot na pole C3? ANO  
b) Na jakém poli robot skončí? G2



✓ 1

12. Vysvětli, k čemu se využívá příkaz if.

K PODMÍNEČNÉMU VYKONÁNÍ PŘÍKAZU

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

X



## Posttest

### Matematika

- Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?  
 4,3krát
- Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o 107°. Kolikrát se motor musí otočit, aby se robot otočil o 45°?  
 215,3°
- Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.  
 A) 3536krát B) 14,1krát
- Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



16 a 26 cm

### Fyzika

- Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je 10kΩ a vstupní napětí je 5 V?  
~~100 mA~~ 0,5 A
- Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?  
 2500 Ω
- Jak poznáš, jak velký má rezistor odpor?  
 MĚŘ POUŽÍ KONVENČNÍ MĚŘNÍ NEBO SE POHÝBAT NA PŘÍKLEP
- Vysvětli fungování fotorezistoru.  
 MĚŘ INTENZITU SVĚTLA NEBO OSVĚLENÍ
- Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

1372 m

13.

10 Ω

}

REZISTOR (EXIT - SUCCESS)

}

## Programování

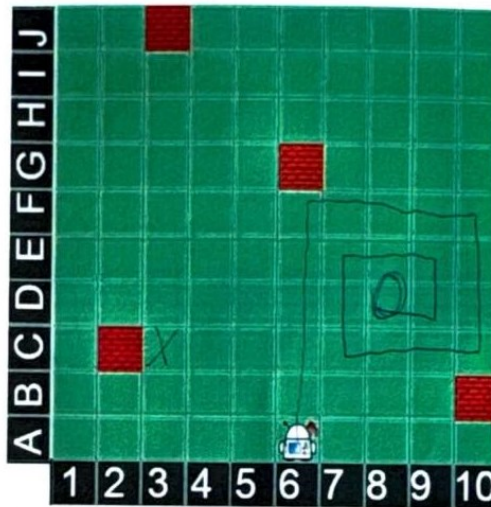
10. Vysvětli pojem algoritmus.

VYŘEŠIT DANÝ TYP PŮVODY.  
PŘESNÝ NÁVOD ČI POSTUP, KTERÝM LZE

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zed), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

a) Dostane se někdy robot na pole C3? NE

b) Na jakém poli robot skončí? D8



12. Vysvětli, k čemu se využívá příkaz if.

PODMÍNĚNĚ VYKONÁNÍ PŘÍKAZU

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

```

INT MAIN (INT ARGV, CHAR** ARGV)
{
    INT CISLA[10] = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
    INT i;
    FOR (i=0; i<10; i++)
    {
        IF (CISLA[i] >= 500 && CISLA[i] <= 10)
            CONTINUE;
        PRINTF ("%d\n", CISLA[i]);
    }
}
    
```

$$0 = 2 \cdot 3,14 \cdot 25 \cdot u = \overset{700}{\dots} \overset{157}{\dots}$$

## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?

✓ 1  $4,458x$

2. Pokud se motor robotického auta otočí 512krát, celý robot změní svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?

✓ 1  $215,327x$

3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

✓ 1 a)  $35\ 35,911x$  b)  $14,143x$

4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



✓ 1 Šířka: 16 cm

délka: 26 cm

### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10\text{ k}\Omega$  a vstupní napětí je 5 V?

0,5  $0,5\text{ A}$

6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?

✓ 1  $2500\ \Omega$

7. Jak poznáš, jak velký má rezistor odpor?

✓ 1 Musím použít konvenční multimeter (vzorec:  $R = \frac{U}{I}$ )

8. Vysvětli fungování fotorezistoru.

✓ 1 ~~Fotorezistor~~ jeho el. odpor se snižuje se zvyšující intenzitou světla.

9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

x  $1372\text{ m}$

11.58

## Programování

10. Vysvětli pojem algoritmus.

Teoretický princip řešení problému

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

a) Dostane se někdy robot na pole C3?

b) Na jakém poli robot skončí?

✓ a) Ne

✓ b) Skončí na  
D8



12. Vysvětli, k čemu se využívá příkaz if.

✓ Podmíněné vykonání kódu

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

✓ (loop) díky tomuto příkazu můžeme jednou se vícekrát opakovat určitý blok kódu opakovat

6,28



$$250 : 36,2 \doteq 6,91$$

$$\frac{5 \cdot 2}{10} = 2,5$$

## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?

X

74 krát

2. Pokud se motor robotického auta otočí 512krát, celý robot změní svůj směr o 107°. Kolikrát se motor musí otočit, aby se robot otočil o 45°?

✓ 1

$$\doteq 215$$

3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

✓ 1

$$a) \doteq 3537,92$$

$$b) \doteq 14,74$$

4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



0,5

$$\text{Podstava: } 76,26$$

$$\text{Výška: } 8$$

### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je 10kΩ a vstupní napětí je 5 V?

✓ 1

$$I = \frac{5}{10000}$$

$$I = 0,0005 \text{ A}$$

6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?

✓ 1

$$R = \frac{5}{0,002}$$

$$R = 2500 \Omega$$

7. Jak poznáš, jak velký má rezistor odpor?

X

8. Vysvětli fungování fotorezistoru.

X

9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

X

$$343 : 2 = 171,5$$

$$\begin{array}{r} 14 \\ 03 \\ 70 \\ 0 \end{array}$$

(5 b)

## Programování

10. Vysvětli pojem algoritmus.

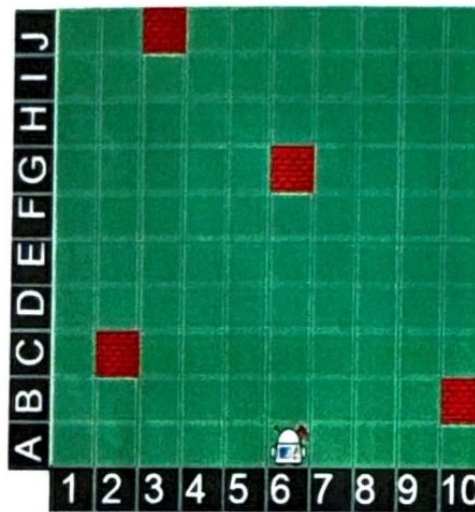
x *VĚC CO SE OPAKUJE POŘÁD DO KOLA*

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zed), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměřován směrem vzhůru.

✓ a) Dostane se někdy robot na pole C3? *NE*

x b) Na jakém poli robot skončí? *F10*

0,5



12. Vysvětli, k čemu se využívá příkaz if

x

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

x

## Posttest

### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?

X Kolo se musí otočit 350krát

2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o  $107^\circ$ . Kolikrát se motor musí otočit, aby se robot otočil o  $45^\circ$ ?

X motor se otočí 64krát

3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

0,15 x a) ~~250~~ 150  
b) 150

512 : 36,2 = 14  
3120 : 36,2 = 85

4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?

✓ 1 Délku má 26 cm a šířku 16 cm  
výšku 8 cm



### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je  $10\text{k}\Omega$  a vstupní napětí je 5 V?

X 500 mA

6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?

X 500 000 mA

7. Jak poznáš, jak velký má rezistor odpor?

X Podle toho jak je velký

8. Vysvětli fungování fotorezistoru.

X Nevím

9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

X Nevím

1,5 f

## Programování

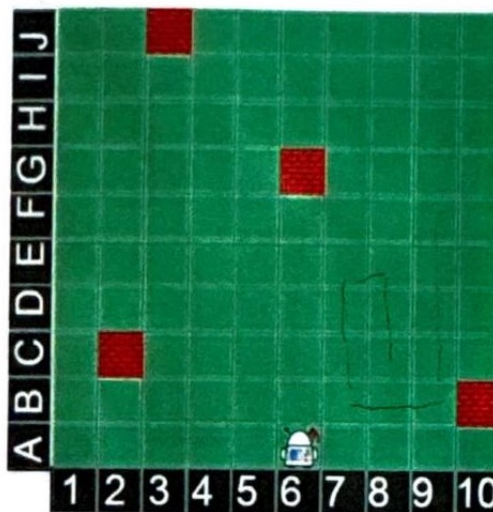
10. Vysvětli pojem algoritmus.

*je to stále opakující se příkaz*

11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

a) Dostane se někdy robot na pole C3? *ne*

b) Na jakém poli robot skončí? *8C*



12. Vysvětli, k čemu se využívá příkaz if.

*k výkonu práce kterou jsme zadaly robotovi*

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

*LOW HiStl*



628

$700 : 157 = 4,45$   
 $157 \text{ cm } 360^\circ$

6 90

### Posttest

#### Matematika

1. Kolo má poloměr 25 cm. Kolikrát se musí otočit, aby ujelo 7 m?

✓ kolo se musí otočit 4,4x  $O = 2 \cdot \pi \cdot r$   $6,28 \cdot 25$

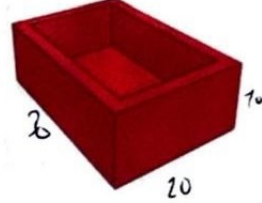
$$\begin{array}{r} 6,28 \\ \times 25 \\ \hline 3140 \\ 1256 \\ \hline 757,00 \end{array}$$
  
4,4  
45  
220

2. Pokud se motor robotického auta otočí 512krát, celý robot změni svůj směr o 107°. Kolikrát se motor musí otočit, aby se robot otočil o 45°?

✓ musí se otočit 198x  $512 : 107 = 4,7$   
3. Krokový motor robotického auta se otočí 512krát a auto ujede 36,2 cm. Kolikrát se musí motor otočit, aby auto ujelo: a) 250 cm b) 1 cm.

✓ 3536x  $512 \cdot 7$   
 $19,14x$

4. Mějme kvádr s rozměry podstavy 20 cm a 30 cm s výškou 10 cm. Chceme z kvádrů vyjmout menší kvádr, tak aby vznikla vana s šířkou stěny 2 cm. Jakou musí mít menší kvádr šířku a jakou délku?



0,5 20 cm a 70 cm s výškou 8 cm

#### Fyzika

5. Jaký proud teče elektrickým obvodem s jedním rezistorem, jestliže jeho odpor je 10kΩ a vstupní napětí je 5 V?

✗  $I = \frac{U}{R} = \frac{5}{10000} = 0,0005 \text{ A}$   
 $500 \mu\text{A}$

6. Jaký odpor musí být v obvodu, aby při vstupním napětí 5 V obvodem protékal proud 2 mA?

✗  $R = \frac{U}{I} = \frac{5}{0,002} = 2500 \Omega$

7. Jak poznáš, jak velký má rezistor odpor?

8. Vysvětli fungování fotorezistoru. Podle počtu rezistorů

9. Urči vzdálenost překážky, jestli se zvuková vlna ozvěnou od zdroje vrátila za 4 s. Počítej s rychlostí zvuku 343 m/s.

✓  $s = \frac{v \cdot t}{2} = \frac{343 \cdot 4}{2} = 686 \text{ m}$

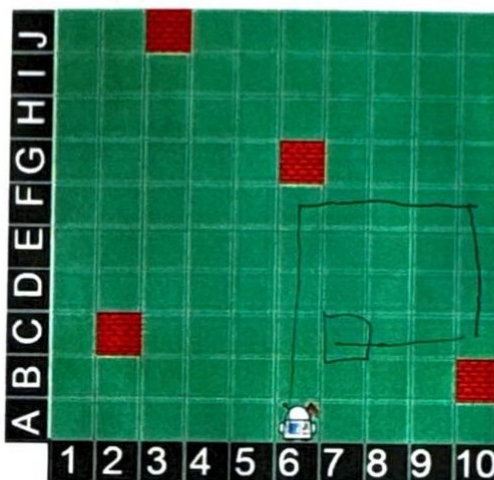
5 b

## Programování

10. Vysvětli pojem algoritmus.

X  
Algoritmus je příkaz který se pořád opakuje  
11. Robot má tři pravidla pro svůj pohyb. Pokud dojde před překážku (zeď), otočí se vpravo, obdobně pokud dojde před pole na, kterém již byl, otočí se vpravo. Třetí pravidlo zní, že pokud je před robotem volné pole, postoupí na něj. Na začátku je robot nasměrován směrem vzhůru.

- 0,5  
a) Dostane se někdy robot na pole C3? Ne ✓  
b) Na jakém poli robot skončí? C7 ✓



12. Vysvětli, k čemu se využívá příkaz if.

X  
k programování

13. Vysvětli příkaz cyklu v nějakém programovacím jazyce.

X  
—