

Příprava výuky programování platformy Arduino pro střední školy

Bc. Viktor Vráblík

Diplomová práce
2018/2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Viktor Vráblík**
Osobní číslo: **A17371**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Učitelství informatiky pro střední školy**
Forma studia: **prezenční**

Téma práce: **Příprava výuky programování platformy Arduino pro střední školy**

Téma anglicky: **Preparation of an Arduino Programming Course for Secondary Schools**

Zásady pro vypracování:

1. Zpracujte literární rešerši na téma využití platformy Arduino pro výuku programování mikropočítačů.
2. Navrhněte obsah výuky vhodný pro střední školy
3. Vytvořte výukové prezentace k jednotlivým tématům výuky.
4. Vypracujte výukové materiály pro studenty a učitele včetně úkolů k procvičení.
5. Vytvořte webové stránky pro podporu výuky a umístěte tam vytvořené materiály.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **BANZI, Massimo. Getting started with Arduino. 2nd ed. Farnham: O'Reilly, 2011. ISBN 9781449309879.**
2. **KALHOUS, Zdeněk a Otto OBST. Školní didaktika. Vyd. 2. Praha: Portál, 2009. ISBN 978-80-7367-571-4.**
3. **MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012, xx, 699 p. ISBN 1449313876.**
4. **PASCH, Marvin. Od vzdělávacího programu k vyučovací hodině. Vyd. 2. Praha: Portál, 1998. ISBN 80-7367-054-2.**
5. **PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd. Praha: BEN – technická literatura, 2004, 159 s. ISBN 80-7300-110-1.**
6. **PRŮCHA, Jan. Moderní pedagogika. Šesté, aktualizované a doplněné vydání. Praha: Portál, 2017. ISBN 978-80-262-1228-7.**

Vedoucí diplomové práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

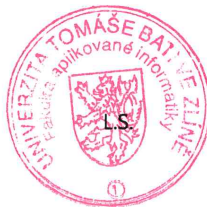
3. prosince 2018

Termín odevzdání diplomové práce:

15. května 2019

Ve Zlíně dne 7. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Mgr. Roman Jašek, Ph.D.
garant oboru

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 16. 5. 2019

Viktor Vráblík, v. r.
podpis diplomanta

ABSTRAKT

Cílem této práce bylo vytvořit materiály pro podporu výuky programování platformy Arduino. Byla vypracována rešerše platformy Arduino se zaměřením na výuku, včetně případných alternativ. Pro zjištění reálného využití na středních školách byl rozeslán dotazník učitelům středních škol. V závěru teoretické části jsou popsány vhodné programovací jazyky se zaměřením na programovací jazyk Arduino.

V rámci praktické části práce byly vytvořeny studijní materiály pro výuku programování platformy Arduino. Dílčí kapitoly vypracovaných materiálů odpovídají společným tématům rámcových vzdělávacích programů pro různé obory středních škol. Materiály byly vypracovány ve dvou variantách. Pro učitele a žáky. Jako závěrečná část práce byl vytvořen webový portál, který obsahuje všechny vypracované materiály, včetně prezentací a ukázkových programů řešených v simulátoru Tinkercad. Tento portál by měl sloužit jako ucelený zdroj materiálů potřebných pro výuku.

Klíčová slova: Arduino, výuka, mikropočítač, programování, algoritmizace

ABSTRACT

Main goal of this work was to create materials to support the teaching of Arduino programming. The Arduino platform has been researched with a focus on teaching, including possible alternatives. A questionnaire was sent out to secondary school teachers to find real use in secondary schools. At the end of the theoretical part are described suitable programming languages with focus on programming language Arduino.

In the practical part of the work, study materials for teaching the Arduino platform were created. Subchapters of the elaborated materials correspond to the common topics of the Framework Education Programme for various fields of secondary education. The materials were developed in two variants. For teachers and pupils. As a final part of the work was created a web portal, which contains all the elaborated materials, including presentations and sample programs solved in the Tinkercad simulator. This portal should serve as a coherent source of teaching materials.

Keywords: Arduino, education, microcomputer, programming, algorithmization

Rád bych poděkoval vedoucímu práce, Ing. Janu Dolinayovi, Ph.D. za jeho odborné rady a pomoc při vypracování této práce. Dále bych chtěl poděkovat Mgr. Iloně Kočvarové, Ph.D. za pomoc při konstrukci dotazníku.

V neposlední řadě bych chtěl poděkovat celé své rodině, za neutuchající podporu při celém mém studiu.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 ARDUINO	11
1.1 HISTORIE	11
1.2 HARDWARE.....	11
1.2.1 Hardwarové komponenty desky	11
1.2.2 Typy desek.....	13
1.2.2.1 Arduino Nano	13
1.2.2.2 Arduino Micro	13
1.2.2.3 Arduino UNO	14
1.2.2.4 Arduino Mega.....	15
1.2.3 Klony	16
1.2.4 Shield	17
1.3 TVORBA PROGRAMU PRO PLATFORMU ARDUINO	18
1.3.1 Arduino IDE	18
1.3.2 Ardublockly	19
1.3.3 Autodesk Tinkercad Simulátor	20
2 ALTERNATIVY K PLATFORMĚ ARDUINO	22
2.1 LEGO MINDSTORMS.....	22
2.2 MICRO:BIT	23
3 ARDUINO NA STŘEDNÍCH ŠKOLÁCH V ČR	26
3.1 DEMOGRAFICKÉ ÚDAJE.....	26
3.2 PROGRAMOVÁNÍ	28
3.3 ARDUINO VE VÝUCE.....	29
3.3.1 Nevyužití.....	29
3.3.2 Využití	30
3.4 SHRUTÍ	31
4 VÝUKA PROGRAMOVÁNÍ	32
4.1 PROGRAMOVACÍ PARADIGMATA	32
4.1.1 Procedurální paradigma	32
4.1.2 Objektově orientované paradigma	33
4.2 VÝBĚR JAZYKA	33
4.2.1 Cílová skupina	33
4.2.2 Aktuálnost.....	33
4.2.3 Dostupnost	34
4.2.4 Syntaxe.....	34
4.2.5 Literatura.....	34
4.3 PLATFORMA ARDUINO A VÝUKA.....	34
II PRAKTICKÁ ČÁST	36
5 RÁMCOVÉ VZDĚLÁVACÍ PROGRAMY	37

5.1	RVP OBECNĚ	37
5.2	RVP GYMNÁZIA	38
5.2.1	RVP – Informatika a informační a komunikační technologie	38
5.2.1.1	Cílové zaměření vzdělávací oblasti	38
5.2.1.2	Vzdělávací obsah	38
5.2.2	ŠVP – Gymnázium Uherské hradiště – Vyšší stupeň vzdělávání	39
5.3	RVP PRŮMYSLOVÁ ŠKOLA	40
5.3.1	RVP	40
5.3.1.1	Vzdělávací obsah	40
5.3.2	ŠVP – Střední průmyslová škola Zlín Technické lyceum	41
5.4	SROVNÁNÍ	42
6	PŘÍPRAVA HODINY	44
6.1	MATERIÁLY PRO UČITELE	44
6.1.1	Didaktická část	44
6.1.1.1	Téma vyučovací hodiny	44
6.1.1.2	Cíl hodiny	44
6.1.1.3	Klíčové kompetence	45
6.1.1.4	Organizační formy výuky	46
6.1.1.5	Metody výuky	46
6.1.1.6	Materiální didaktické prostředky	47
6.1.2	Výuka	47
6.1.2.1	Teorie	47
6.1.2.2	Příklady	47
6.1.2.3	Úkoly	48
6.2	MATERIÁLY PRO ŽÁKY	48
7	ZPRACOVANÁ TÉMATA	49
7.1	ALGORITMIZACE	50
7.1.1	Úvod	50
7.1.2	Základní struktura a zápis	50
7.1.3	Cyklus	50
7.1.4	Pole	50
7.1.5	Procvičování	51
7.2	ARDUINO – ÚVOD	51
7.2.1	Úvod	51
7.2.2	Zápis kódu a struktura programu	51
7.2.3	Vstupy a výstupy	52
7.2.4	Proměnné	52
7.2.5	Podmínky	52
7.2.6	Cykly	52
7.2.7	Pole	52
7.3	ARDUINO – ZÁKLADNÍ HW PRVKY	53
7.3.1	Sedmi segmentový displej	53
7.3.2	Ultrazvukový měřič vzdálenosti, funkce	53
7.3.3	Servo, PWM	53
7.3.4	Piezo bzučák	53
7.3.5	Teplotní senzor	53

7.4	ARDUINO – POKROČILÉ HW PRVKY.....	54
7.4.1	LCD displej.....	54
7.4.2	Tlačítková matice.....	54
7.4.3	PIR senzor.....	54
7.4.4	LED pásy	54
7.5	ARDUINO – ZÁVĚREČNÝ PROJEKT.....	54
8	PREZENTACE GOOGLE	56
9	WEBY GOOGLE.....	57
9.1	EDITOR	57
9.2	PUBLIKOVÁNÍ	58
	ZÁVĚR.....	59
	SEZNAM POUŽITÉ LITERATURY	60
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	64
	SEZNAM OBRÁZKŮ.....	65
	SEZNAM TABULEK	66
	SEZNAM PŘÍLOH	67

ÚVOD

Arduino je zařízení spojující abstraktní programové prostředí s fyzickým světem elektro-technických komponent. Vzniklo jako prostředek pro výuku studentů, kteří nemají příliš zkušeností s programováním nebo elektrotechnikou. Z toho důvodu je i samotná obsluha a zápis kódu velmi jednoduchá záležitost, i pro mladší uživatele. Arduino je svou podstatou velmi variabilní zařízení využitelné v širokém počtu různých odvětví nejen v rámci výuky.

Platforma Arduino má díky své popularitě velmi širokou uživatelskou základnu, díky tomu i velký počet dostupných návodů a tutoriálů i celých webů dedikovaných tomuto projektu. Na druhou stranu je velmi složité najít didakticky správně zpracované přípravy výuky zaměřené na toto téma. Pokud ano, tak jsou spíše zaměřené na elektrotechnickou část práce, nikoli na samotnou algoritmizaci.

Práce má za cíl vytvořit ucelenou přípravu výuky programování s využitím platformy Arduino, včetně zpracovaných podpůrných materiálů jako jsou prezentace a webový portál jako podporu výuky. Díky své rozmanitosti je Arduino využitelné nejen jako prostředek pro výuku programování mikropočítačů, ale s vhodným technickým vybavením i jako alternativa ke klasické výuce programování. Poslední zmiňovaná možnost byla hlavní myšlenkou celé této práce.

Teoretická část je věnována obecnému náhledu na platformu Arduino. Je nastíněna geneze vzniku, základní typy desek a způsoby obsluhy. V teoretické rovině je popsána i výuka programování, hlavně výběr programovacího jazyku a paradigmatu, se zaměřením právě na platformu Arduino. Pro praktickou část bylo vypracováno dotazníkové šetření zaměřené na učitele středních škol. V praktické části jsou zkoumány RVP různých středních škol, a výběr témat ke zpracování. Dále je popsána základní struktura výukových materiálů a rozvržení navrhované výuky. Poslední část je věnována způsobu vytvoření prezentací a webové stránky.

I. TEORETICKÁ ČÁST

1 ARDUINO

Arduino je platforma propojující reálný fyzický svět (senzory, tlačítka, LED...) a svět výpočetní techniky. Vznikla jako školní projekt studentů pro studenty. Díky své jednoduchosti a univerzálnosti se velmi rychle rozšířila i mezi širokou veřejností. V dnešní době je platforma Arduino jedna z předních zástupců v oblasti mikropočítačů, potažmo mikrokontrolerů.

1.1 Historie

Celý projekt byl založen na práci studenta jménem Hernando Barragán, který ve své diplomové práci *Wiring: Prototyping Physical Interaction Design* vytvořil v roce 2004 základ knihovny *Wiring*, která se později stala základem pro programovací jazyk Arduino. Celou práci vedl Massimo Banzí. [1]

Projekt Arduino byl odvozen z výše uvedené diplomové práce. Za celým tímto projektem stál již zmíněný vedoucí původní práce Massimo Banzí a další vědci jako byl David Cuartielles, Tom Igoe, Gianluca Martino a David Mellis. Tento projekt vyvinuly na institutu *Interaction Design* v italském městě Ivrea na severu Itálie. Na oficiálních stránkách pouze uvádí, že základem byl projekt *Wiring*. Využití diplomové práce, bez přiznání dostatečných zásluh byla předmětem sporů, mezi autorem původní práce a týmem stojícím za platformou Arduino. [2; 3]

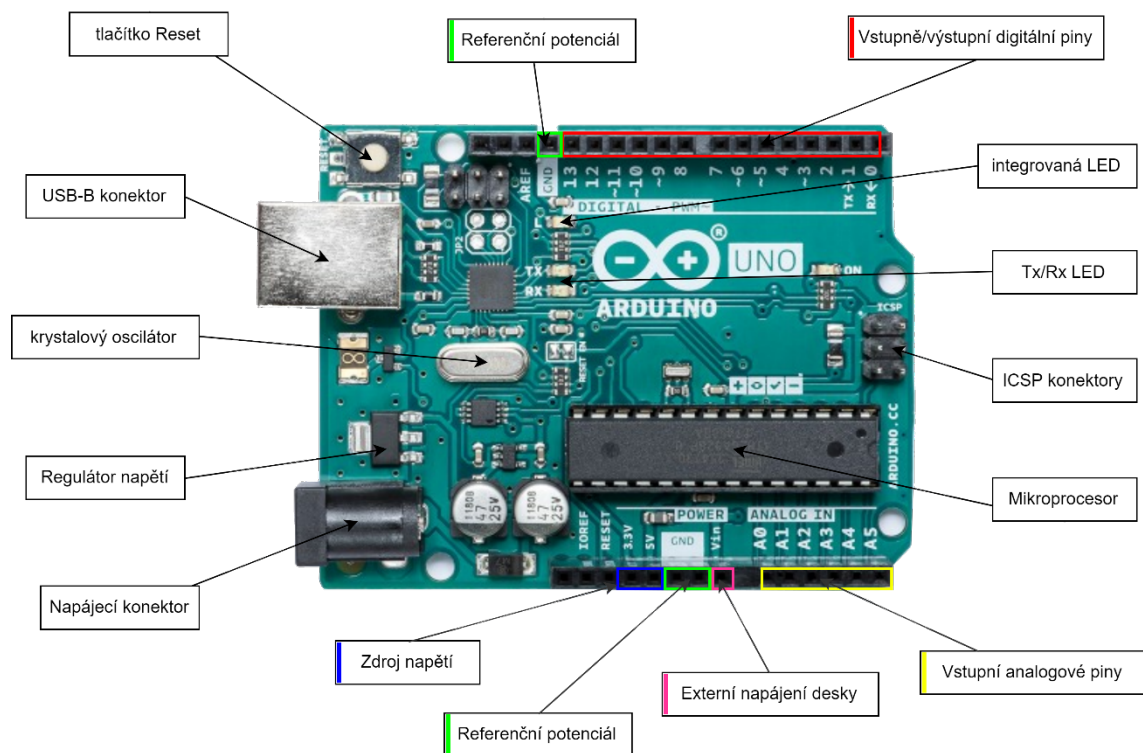
První deska Arduino byla veřejnosti představena v roce 2005 jako nástroj pro studenty, kteří neměli žádné, nebo velmi malé, zkušenosti v oblasti elektrotechniky a programování mikropočítačů. V dnešní době je používána nejenom jednotlivci ale i velkými korporacemi. [4]

1.2 Hardware

V následující kapitole jsou popsány hlavní hardwarové komponenty na vývojové desce, stejně tak i základní typy desek a hardwarová rozšíření.

1.2.1 Hardwarové komponenty desky

Na desce se nachází spousta různých hardwarových prvků. Různé verze desek, nemusí obsahovat všechny uvedené prvky (demonstrováno na desce Arduino UNO), případně se mohou lišit v umístění, základ však zůstává velmi podobný.



Obrázek 1 Schéma umístění základních prvků na desce Arduino UNO

Výpočetní jádro desky tvoří osmibitový RISC mikročítač z rodiny AVR mikroprocesorů společnosti Atmel. Pro komunikaci je deska osazena konektorem USB-B, který slouží zároveň i jako jedna z možných variant napájení. Programování desky lze provádět i pomocí ICSP (In-Circuit Serial Programming) protokolu. Pro napájení je mimo USB-B možné využít i napájecí jack konektor 2.1 mm, nebo externí napájecí pin „Vin“ (na obrázku označení růžově). Zdroj referenčního potenciálu je realizován pomocí tří pinů „GND“ (na obrázku vyznačeny zeleně). Pracovní napájení většiny desek je 5 V, regulátor napětí však umožňuje napájet desku až 12 V (vždy je nutné ověřit si tuto informaci pro konkrétní použitou desku).

Pro připojení akčních prvků jsou na desce vyvedeny vstupně/výstupní piny. Konkrétně se jedná o digitální vstupně/výstupní piny (na obrázku vyznačeny červeně) a analogové vstupní piny (na obrázku vyznačeny žlutě). Analogový výstup je realizován pomocí PWM (Pulse Width Modulation – pulsně šířková modulace) na dedikovaných digitálních pinech, označených vlnovkou. Pro sériovou komunikaci jsou vyhrazeny piny 1 a 0, označeny TX a RX. Napájení akčních prvků je prováděno pomocí pinů zdroje napětí 5 V nebo 3,3 V (na obrázku vyznačeno modře).

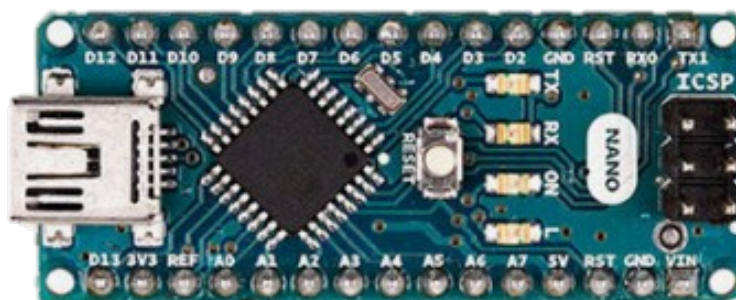
1.2.2 Typy desek

Pro širokou škálu projektů je vyžadováno jiných konstrukčních řešení samotné desky. V některých případech je zapotřebí menší velikosti na úkor počtu vstupně výstupních pinů, v jiných projektech je vyžadováno připojit více zařízení a na velikosti desky příliš nezáleží. Proto během let vzniklo několik oficiálních typů, které pokrývají téměř všechny potřebné konfigurace.

V následující části jsou uvedeny čtyři základní typy desek, Arduino Nano, Arduino Micro, Arduino UNO a Arduino Mega. Všechny pochází z oficiální distribuce firmy Arduino.

1.2.2.1 Arduino Nano

Nejmenší ze všech jmenovaných desek je Arduino Nano. Rozměr desky je pouhých 18 mm x 45 mm a hmotnost přibližně 7 g. Deska je řízena mikroprocesorem ATmega328 o taktovací frekvenci 16 MHz. [5]

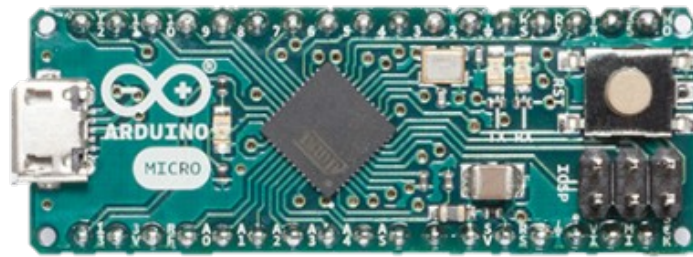


Obrázek 2 Arduino Nano [5]

Uživatel má k dispozici 22 digitálních vstupně/výstupních pinů (z toho 6 je schopno pracovat s PWM) a 6 analogových vstupních pinů. Celková flash paměť je 32 KB, z toho je pro samotný program dostupných 30 KB. Oproti větším deskám neobsahuje standardní USB-B konektor, ten je nahrazen konektorem Mini-B. Stejně tak není přítomen ani napájecí jack konektor. Svou konstrukcí je určen převážně jako část plošného spoje, nikoli pro práci na prototypu. [5]

1.2.2.2 Arduino Micro

Další z řady desek je Arduino Micro. Rozměry desky jsou srovnatelné s deskou Arduino Nano, 18 mm x 48 mm. Váha je oproti předchozímu modelu téměř dvojnásobná, 13 g. Na první pohled by se mohla jevit jako jen upravená verze desky Arduino Nano, ale není tomu tak. Tato deska byla vyvíjena ve spolupráci s americkou společností Adafruit. [6]



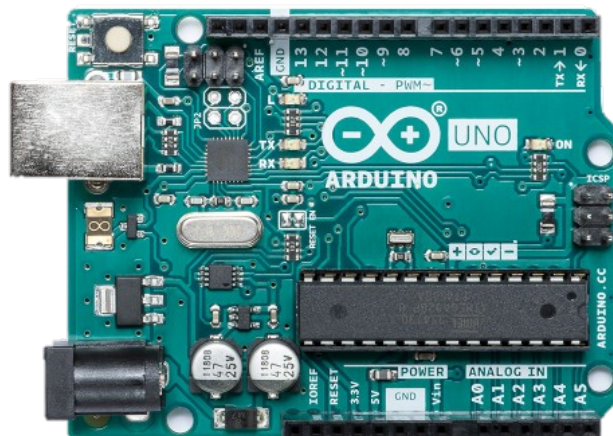
Obrázek 3 Arduino Micro [6]

Jádro desky tvoří procesor ATmega32U4, s taktovací frekvencí 16 MHz. Pro program je na desce dostupných 28 KB, z celkových 32 KB. Pro zapojení je dostupných 20 vstupně/výstupních digitálních pinů (z toho 7 podporujících PWM) a 12 analogových vstupních pinů. [6]

Procesor řídící Arduino Micro umožňuje, mimo klasické využití, registrovat připojenou desku jako myš nebo klávesnici, pomocí třídy USB HID (human interface device). [6]

1.2.2.3 *Arduino UNO*

Pravděpodobně nejoblíbenější deska, která poskytuje dostatečný počet vstupně/výstupních pinů pro potřeby běžně velkých projektů případně výuky.



Obrázek 4 Arduino UNO [7]

Deska je řízena procesorem ATmega328P o frekvenci 16 MHz. Velikost programu je omezena na 31,5 KB z celkových 32 KB flash paměti. Uživatel má k dispozici 14 digitálních vstupně/výstupních pinů (z toho 6 je možné použít jako zdroje PWM) a 6 analogových vstupních pinů. Velikost samotné desky je přibližně 69 mm x 54 mm. Hmotnost 25 g.

Deska obsahuje standardní USB-B konektor pro připojení k PC a napájecí 2,1 mm jack konektor.

1.2.2.4 *Arduino Mega*

Tato deska je určena primárně pro projekty, které potřebují pro svůj chod velký počet vstupně/výstupních pinů. Může se jednat o ovládání 3D tiskáren, 2D plotterů nebo fréz. Je však možné ji využít pro méně náročné projekty, jelikož poskytuje stejnou funkčnost jako menší desky.



Obrázek 5 Arduino Mega [8]

Desku pohání osmi bitový procesor ATmega2560 firmy Atmel, o taktovací frekvenci 16 MHz. Pro připojení externích zařízení nabízí 54 digitálních vstupně/výstupních pinů (z nichž 15 je možné použít jako výstupy PWM) a 16 analogových vstupů. Deska je osazena flash pamětí o velikosti 256 KB, kde pro program je možné využít prostoru 248 KB. [8]

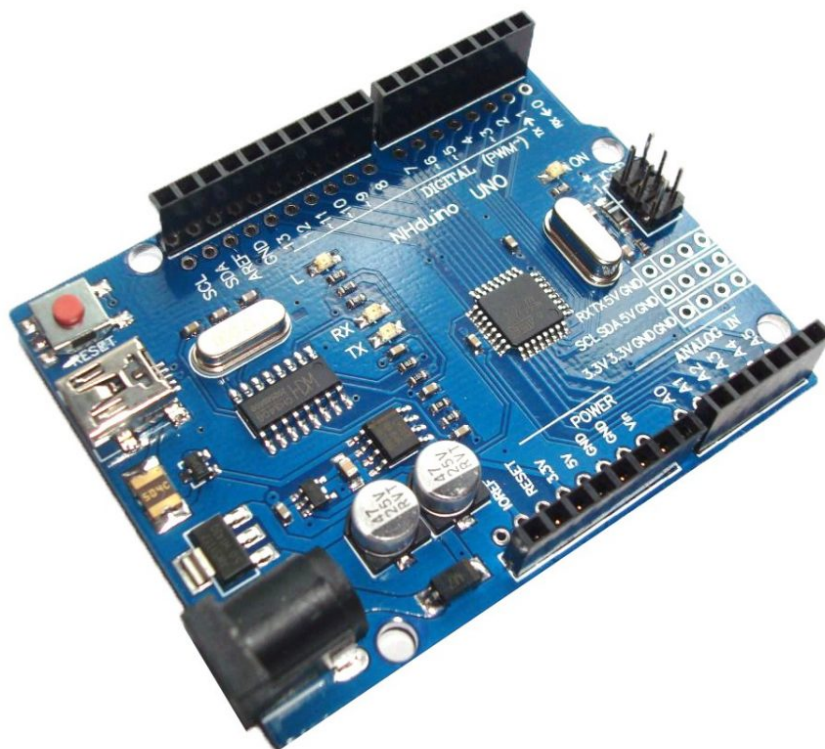
V následující tabulce jsou pro srovnání uvedeny základní vlastnosti všech čtyřech základních typů desek.

Tabulka 1 Srovnání jednotlivých Arduino desek

	Arduino Nano	Arduino Micro	Arduino Uno	Arduino Mega
rozměry (š x v)	45 x 18 mm	48 x 18 mm	69 x 54 mm	102 x 54 mm
váha	7 g	13 g	25 g	37 g
mikroprocesor	ATmega328	ATmega32U4	ATmega328P	ATmega2560
frekvence	16 MHz	16 MHz	16 MHz	16 MHz
digitální I/O	22	20	14	54
PWM	6	7	6	15
analogové I	8	12	6	16
paměť programu	30KB/32KB	28KB/32KB	31.5KB/32KB	248KB/256KB
operační napětí	5V	5V	5 V (max 20 V)	5 V (max 20 V)

1.2.3 Klony

Jelikož je celý projekt Arduino open-source a všechna schémata jsou volně dostupná, chopila se této možnosti spousta dalších výrobců. Převážně tedy čínských firem, které postupně zaplnili trh svými výrobky. V tomto případě se však nedá přímo bavit o kopiích, jsou to pouze výrobky jiných firem, proto jsou obecně uváděny jako kompatibilní klony.



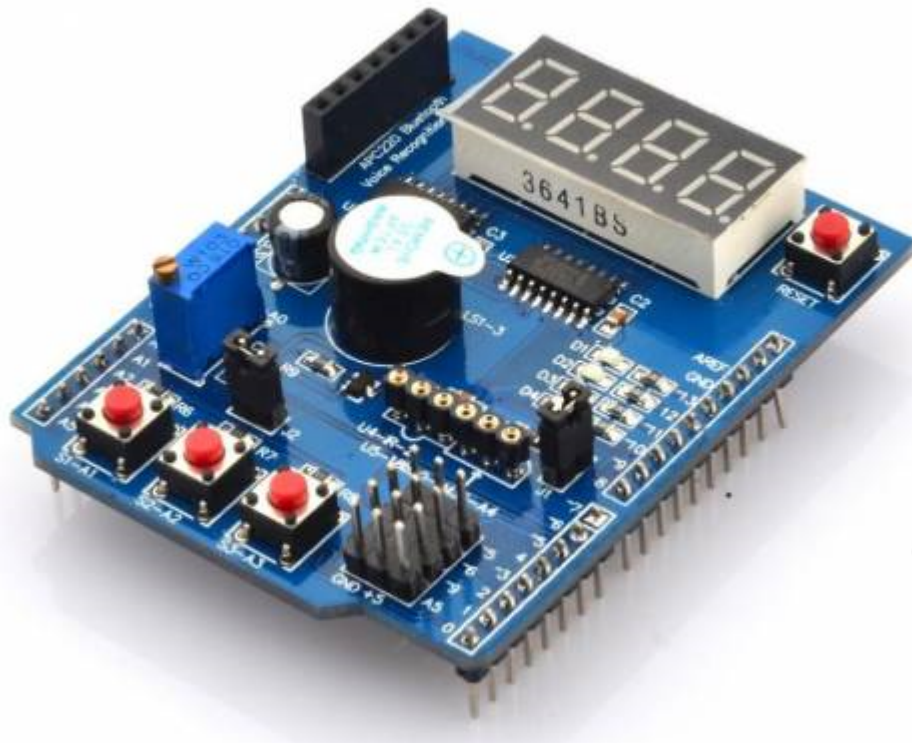
Obrázek 6 Arduino UNO klon [9]

Kvalitou jsou většinou srovnatelné s originálními výrobky společnosti Arduino, ovšem hlavní předností je jejich cena. Ta je oproti originálům opravdu zlomková. Pokud se budeme bavit o konkrétní ceně vývojové desky Arduino UNO. Cena originálu se pohybuje okolo 700Kč. V případě klonů, je cena přibližně tři krát nižší, tedy kolem 200kč (listopad 2018). [9]

Pokud bychom například chtěli vybavit celou jednu třídu, přibližně 30 žáků byla by cena přibližně o 15 000Kč levnější. I za předpokladu, že bychom vzali do úvahu větší poruchovost, je tento rozdíl opravdu markantní. Do této ceny navíc není zahrnuta cena základního příslušenství, jako jsou LED, tlačítka, potenciometry a podobně, na které by tedy v rozpočtu zbylo více peněz.

1.2.4 Shield

Jedná se hardwarové rozšíření desky. Připojení je většinou realizováno samčími konektory, které se shora nasunou na samičí konektory přímo na desce. Odpadá tak komplikace při připojování jednotlivých komponent, například na nepájivém poli pomocí propojovacích vodičů.



Obrázek 7 shield školní rozšíření pro arduino [10]

Vždy jsou určené pro konkrétní rozšíření funkcionality desky. Může se jednat například o Shield s LCD displejem, led maticí, nebo s ethernetovým konektorem.

Stejně tak existují i Shildy vytvořeny primárně pro výuku (na obrázku výše). Obvykle obsahují základní vstupně/výstupní prvky, jako jsou tlačítka, LED, potenciometr, bzučák, ale i prostor pro připojení dalších komponent, například modelářská serva, teplotní a vlhkostní senzory, RFID čtečka a jiné.

Hlavní nevýhodou je fakt, že je uživatel omezen většinou pouze na jeden shield, který může na desku připojit. Existují i zapojení, které umožňují zapojit více shieldů postupně na sebe, deska však tímto velmi roste na rozměrech.

1.3 Tvorba programu pro platformu Arduino

V následující kapitole jsou popsány způsoby programování samotné desky, stejně tak i vývojové prostředí, které je možné využít.

Programovat desku Arduino lze dvěma způsoby (v případě, že nebudeme započítávat možnost zápisu kódu v assembleru). Tím je zápis programu v programovacím jazyku Arduino, nebo grafické programování.

V případě grafického programování je zápis kódu zredukován pouze na postupné skládání funkčních bloků do výsledného programu.

Hlavní výhodou je uživatelská přívětivost. Syntaktická chyba prakticky není možná, jelikož funkční bloky ze své podstaty nelze připojit na syntakticky nesprávné místo. Navíc je samotný kód přehlednější. Tento způsob je určen převážně mladším dětem, nebo méně zkušeným programátorům, kteří si tímto způsobem mohou velmi snadno naprogramovat svůj první program.

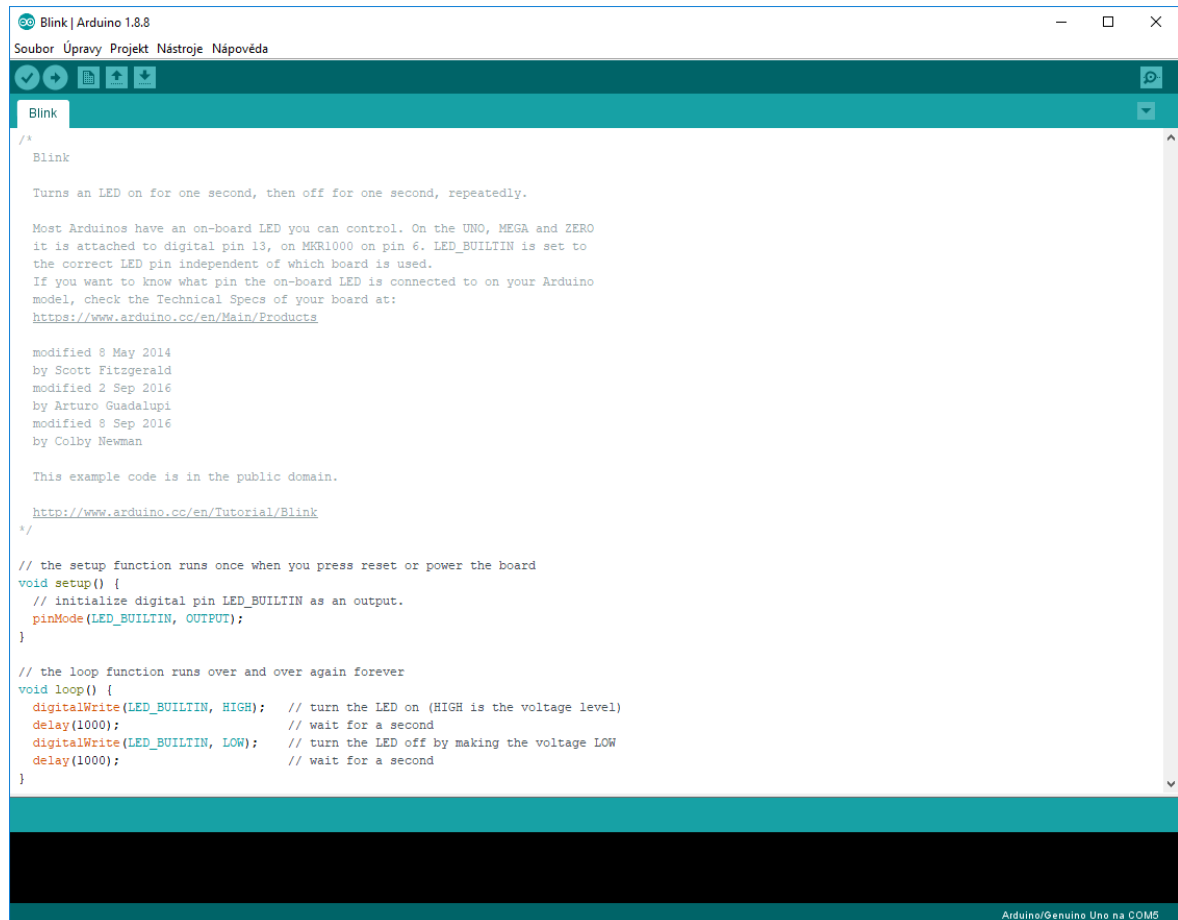
Tento způsob programování bohužel není umožněn ve vývojovém prostředí společnosti Arduino. Je tedy zapotřebí využít aplikací třetích stran. Některá vývojová prostředí jsou uvedena v další kapitole.

Programovací jazyk Arduino vychází z knihovny Wire a je založen na jazyku C/C++. Tento způsob zápisu programu je určen pro pokročilejší uživatele, i když svou jednoduchostí není velkou překážkou i začínajícím programátorům.

Často označované jako IDE, z anglického Integrated Development Environment, je komplexní nástroj pro programování. V další části jsou zmíněny některé nástroje, které lze pro programování využít.

1.3.1 Arduino IDE

Základní programovací prostředí, spravované přímo společností Arduino, je Arduino IDE. Nabízí všechny nástroje pro základní programování. Hlavní okno pro zápis kódu, správu knihoven a sériový monitor, pro komunikaci po sériové lince. Uživateli nabízí i spoustu ukázkových programů, vytvořených přímo společností Arduino.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.8". Below the title bar is a menu bar with "Soubor", "Úpravy", "Projekt", "Nástroje", and "Nápověda". The main workspace contains the "Blink" example code. The code is a C++ program that blinks an LED. It includes a header comment explaining the code's purpose and history, followed by the setup and loop functions. The setup function initializes the LED pin as an output. The loop function turns the LED on for one second and then off for one second, repeating this process forever. The IDE status bar at the bottom indicates "Arduino/Genuino Uno na COM5".

```
/*  
 * Blink  
 *  
 * Turns an LED on for one second, then off for one second, repeatedly.  
 *  
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO  
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to  
 * the correct LED pin independent of which board is used.  
 * If you want to know what pin the on-board LED is connected to on your Arduino  
 * model, check the Technical Specs of your board at:  
 * https://www.arduino.cc/en/Main/Products  
 *  
 * modified 8 May 2014  
 * by Scott Fitzgerald  
 * modified 2 Sep 2016  
 * by Arturo Guadalupi  
 * modified 8 Sep 2016  
 * by Colby Newman  
 *  
 * This example code is in the public domain.  
 *  
 * http://www.arduino.cc/en/Tutorial/Blink  
 */  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

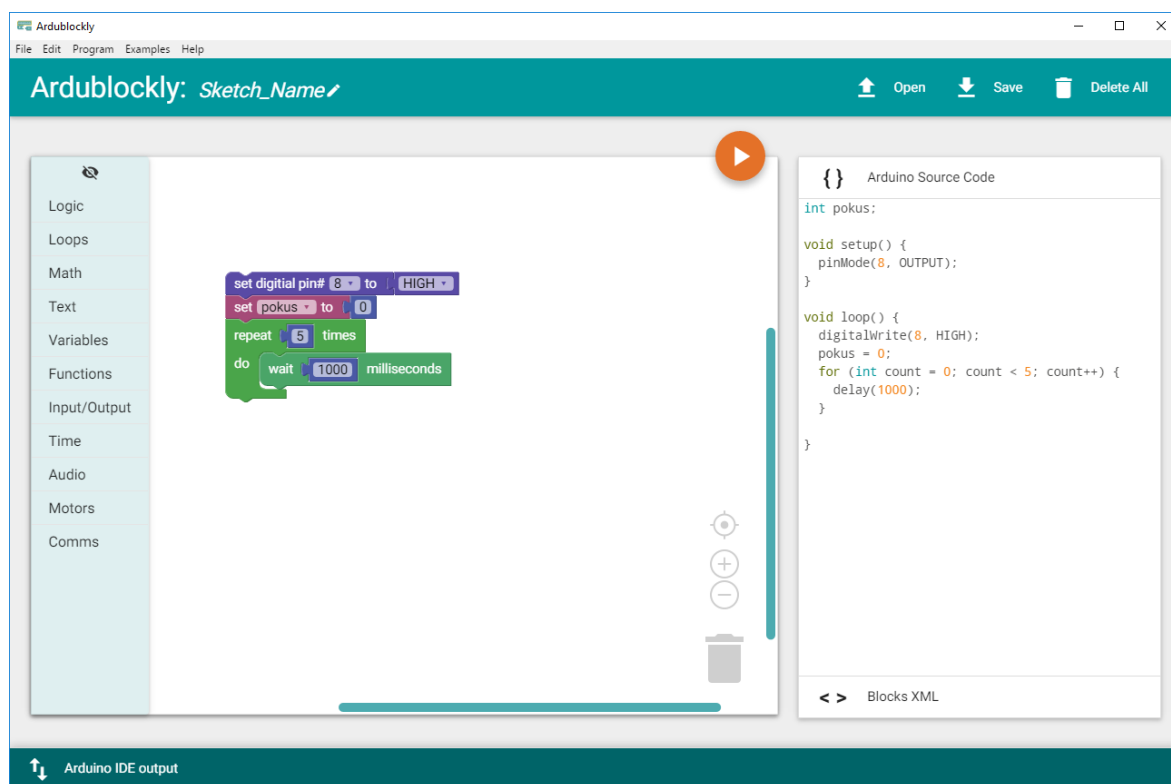
Obrázek 8 Arduino IDE

Dostupná ke stažení ze stránek: <https://www.arduino.cc/en/Main/Software>, pro všechny velké operační systémy, Windows, Linux a MacOS. Pro Windows navíc i ve verzi portable. Na stránkách výrobce je možné využít i online verzi tohoto nástroje. Poskytuje všechny funkce, které nabízí desktopová aplikace, ale umožňuje uživateli využívat prostor pro cloudové ukládání svých projektů. Tyto projekty má tak k dispozici kdekoliv, kde je připojen k internetu.

1.3.2 Ardublockly

Projekt ardublockly se zaměřuje především na programování v grafickém prostředí „bločků“. Dostupný ze stránky: <https://github.com/carlosperate/ardublockly>. Vyvíjen jako nástavba projektu Blockly od firmy Google, upravený pro programování Arduina. Pro

správnou funkci je nutné mít nainstalované Arduino IDE. Na stránce projektu je uveden i odkaz na wiki stránku, která slouží jako obsáhlý návod k obsluze.



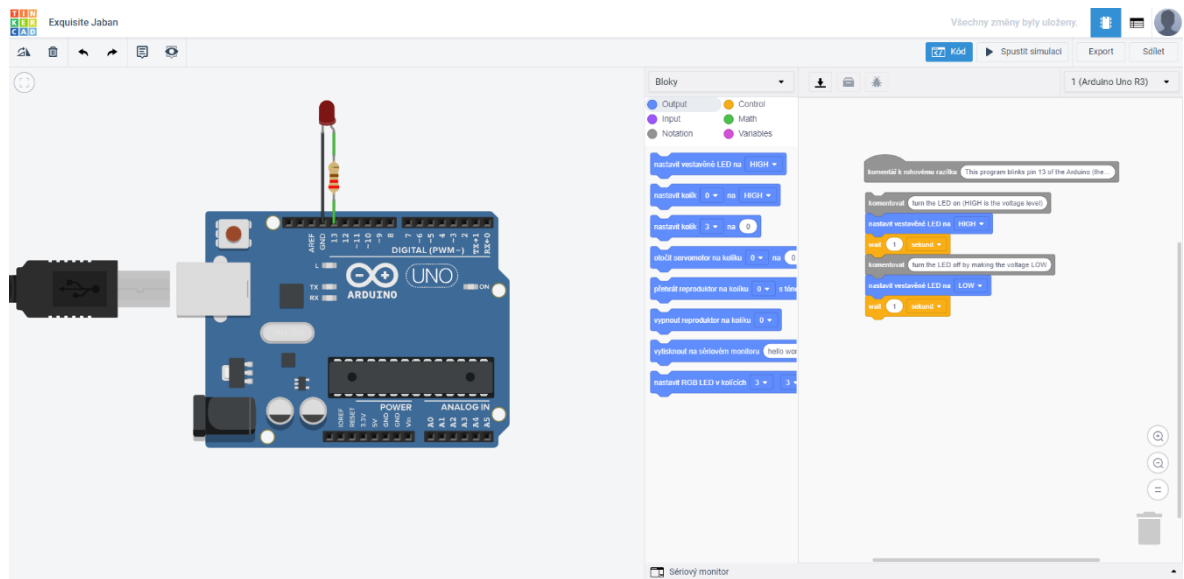
Obrázek 9 Ardublockly

Výsledný kód je zobrazován na pravé straně, v reálném čase podle návrhu v grafickém okně nalevo. Ardublockly umožňuje nahrát kód přímo do vývojové desky, není tedy nutné vygenerovaný kód za účelem nahrání kopírovat do jiného prostředí.

Hlavní nevýhodou je fakt, že nástroj je stále ve vývoji, není tedy dokončená implementace některých pokročilejších funkcí. Další nevýhodou je nutnost znalosti anglického jazyka, jelikož projekt ještě nebyl, a pravděpodobně ani nebude, přeložen do češtiny. Ovšem i přes své drobné nedostatky je jako celek použitelný.

1.3.3 Autodesk Tinkercad Simulátor

Vývojové prostředí firmy Autocad nabízí oproti dříve zmiňovaným navíc ještě jednu zásadní funkci. Simulátor zapojení v reálném čase. Je tedy možné nasimulovat nejen kód, ale i samotnou funkčnost obvodu. V nabídce podporovaných komponent se nachází všechny základní prvky, jako jsou LED, rezistory, spínače, potenciometry a další. Najdeme zde však navíc ještě spoustu dalších pokročilejších součástek jako jsou mikročipy, displeje, tranzistory a podobně.



Obrázek 10 Autodesk Tinkercad

Samotný zápis kódu je možné realizovat v obou variantách. Zápis pomocí grafického editoru, tak i zápis přímo v kódu.

Simulátor má omezenou nabídku hardwarových komponent, ale pro menší, až středně velké projekty je svým vybavením dostačující.

2 ALTERNATIVY K PLATFORMĚ ARDUINO

I přes to že je Arduino pravděpodobně nejrozšířenějším zařízením ve své oblasti, najdou se i jiná řešení poskytující stejnou, nebo velmi podobnou funkcionalitu. V další části této kapitoly jsou uvedeny jen některá z nich. Při výběru byla zohledněna převážně možnost využití ve výuce.

2.1 Lego Mindstorms

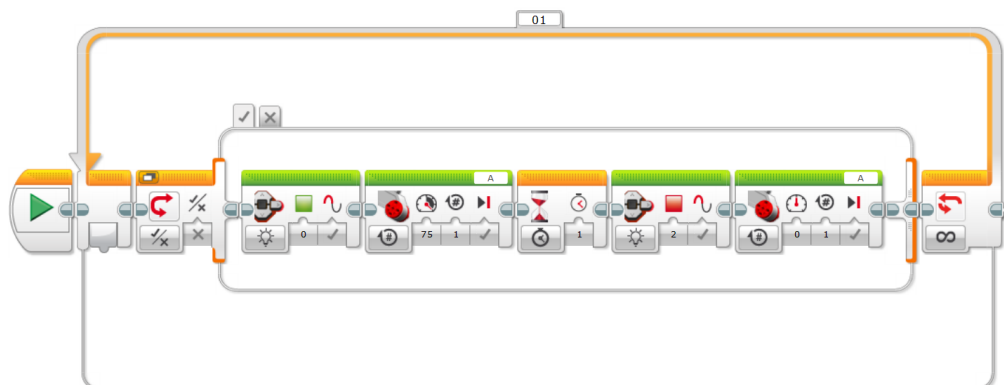
Lego Mindstorms je projekt firmy Lego spojující stavebnici Lego technik a akční prvky jako jsou motory a senzory. Řízení je realizováno pomocí řídicí kostky. Tato stavebnice je primárně určena pro tvorbu robotů a robotických zařízení.



Obrázek 11 LEGO Mindstorms – education [11]

Samotná základní sada nabízí mimo akční elektronické prvky i základní sadu LEGO dílů, dostačující pro stavbu pokročilejších modelů. Navíc je dostupné množství návodů na výrobu konkrétních robotů, určené přímo pro práci s touto sadou.

Programování je realizováno v desktopovém programu LEGO Mindstorms. Samotný zápis kódu poté v grafickém editoru, pomocí funkčních bloků z nabídky. V případě složitějších zapojení je možné vyvářet vlastní funkční bloky.



Obrázek 12 zápis algoritmu v programu Lego Mindstorms

Pro programování lze využít i aplikaci EV3 Programmer dostupnou na chytré zařízení Android nebo IOS. Těmito zařízeními je možné výsledné zapojení i řídit s využitím aplikace Robot Commander. [12]

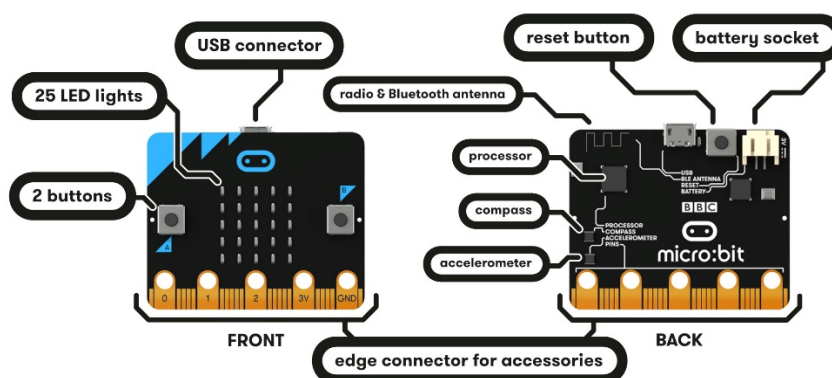
Nevýhodou je hlavně relativně malá variabilita akčních členů. V oficiální nabídce firmy LEGO jsou na výběr pouze dva typy motorů, velký a střední, tlačítko, infračervený senzor vzdálenosti, ultrazvukový senzor vzdálenosti, senzor barvy a intenzity odraženého světla a senzor náklonu. K sadě je pak možné dokoupit i další senzory, jako teplotní senzor nebo senzor zvuku. Existují však i komponenty, které nepocházejí z oficiální distribuce firmy LEGO, ale jsou plně kompatibilní. Například kamera, numerická klávesnice a mnoho dalších. [13; 12]

Hlavní výhodou (která je částečně i nevýhodou) je uzavřenost systému. Tím je zaručena 100% kompatibilita a funkčnost celé stavebnice.

Cena základní sady je 11 590 Kč (duben 2019) která je při srovnání s pořizovací cenou sady Arduino několikanásobná. Pokud by měla být kompletně vybavena celá třída (i za předpokladu jedné sady do dvojice studentů) jednalo by se o poměrně velkou investici v rozpočtu školy. Na druhou stranu není v základu nutné k sadě pořizovat další komponenty. [11]

2.2 Micro:bit

Micro:bit Educational Foundation je nezisková organizace oficiálně založena v roce 2016, jako duchovní nástupce společnosti British Broadcasting Corporation Microcomputer System. Platforma Micro:bit vznikla přímo pro potřeby výuky na středních školách. [14]

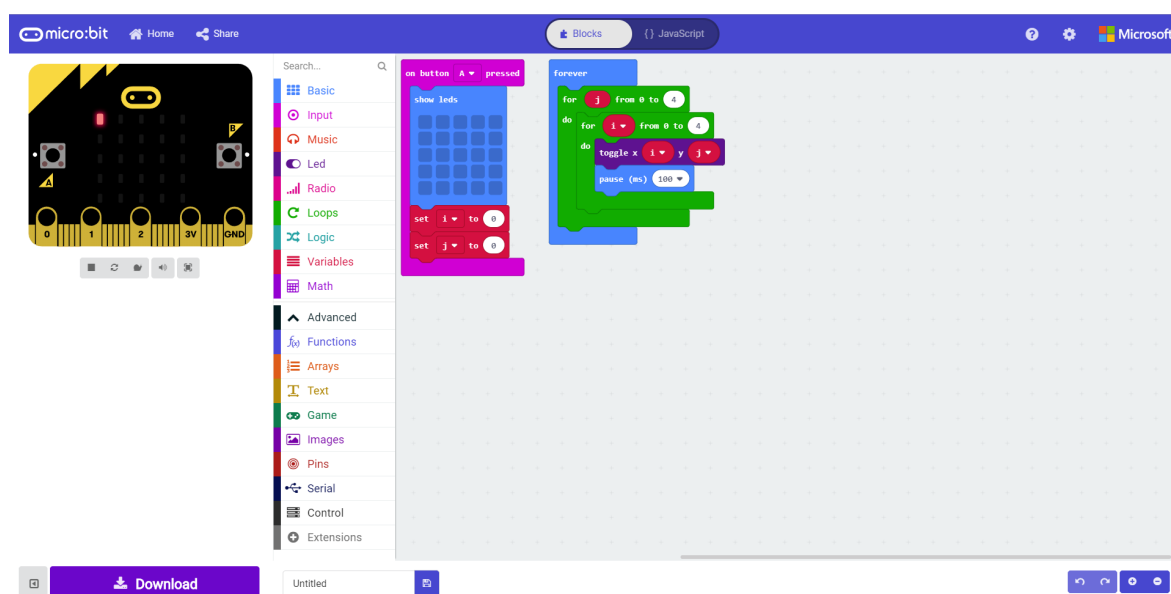


Obrázek 13 Micro:bit přehled komponent [15]

I přes malý rozměr se na desku podařilo umístit dostatečně velké množství akčních prvků. Na přední straně nalezneme dvě tlačítka a matici 25 LED, kterou je možné využít i jako základní detektor světla. Na zadní straně se nachází akcelerometr, kompas, a tlačítko reset. Deska disponuje i Bluetooth anténou i radiovou anténou, která umožňuje komunikovat mezi dvěma Micro:bit deskami. Procesor řídící celé zařízení je navíc vybaven teplotním senzorem, který je také možno využít jako akční člen zapojení. [15]

Na spodní straně nalezneme tři kontakty pro připojení externích zařízení, 3V napájecí kontakt a uzemnění zapojení. Připojení k počítači je realizováno pomocí micro-USB konektoru.

Pro programování jsou dostupné dvě internetové aplikace. Jedna umožňuje programovat Micro:bit pomocí jazyku python. Druhá nabízí grafický editor funkčních bloků i zápis v jazyku java. V tomto editoru je možné i simulovat základní program v reálném čase.



Obrázek 14 zápis algoritmu v aplikaci MakeCode Editor

Cena jedné sady včetně základních komponent je 1399 Kč (duben 2019) což je cena srovnatelná se základní sadou Arduino UNO. [16]

Hlavní nevýhodou je počet vstupně výstupních pinů, které neumožňují připojení většího množství externích komponent. Na druhou stranu je deska už v základu osazena relativně velkým počtem vstupně výstupních prvků, což na základní i středně pokročilé úlohy stačí.

3 ARDUINO NA STŘEDNÍCH ŠKOLÁCH V ČR

Reálné využití na středních školách v České republice, bylo zkoumáno pomocí krátkého dotazníku. Dotazník byl určen pro středoškolské učitele informatiky (případně učitele obdobných předmětů, nebo středoškolských kroužků). Rozesílání probíhalo skrze veřejně dostupné adresy škol (dostupné z <http://stistko.uiv.cz/registr/vybskolrn.asp>). Podle počtu vrácených emailových zpráv, je možné tvrdit, že samotný registr není příliš aktuální.

Dotazník není základem této diplomové práce, slouží pro potvrzení (vyvrácení) některých počátečních předpokladů. Část zjištění vyplívající z dotazníku, byla promítnuta do návrhu struktury zpracovaných materiálů.

Dotazník byl na základě několika předpokladů:

- Platforma Arduino je obecně známá mezi učiteli informatiky.
- Platforma Arduino je využívána spíše na předměty zaměřené na mikropočítače.
- Platforma Arduino je na středních školách aktivně využita ve výuce.
- Základní důvod nevyužití platformy Arduino je její cena.

V rámci dotazníku byly respondenti dotazováni i na některé otázky související s obecně výukou programování.

Úplné znění dotazníku je přiloženo jako příloha (P I – Dotazník). Celý seznam odpovědí jako příloha v souboru příloh v elektronické podobě. Emailové adresy, na které bylo přislíbeno zaslání hotových materiálů, byly z odevzdaného souboru smazány.

Dotazník byl vyhotoven pomocí webové aplikace společnosti Google. Tato aplikace nenabízí podrobné statistické vyhodnocení odpovědí, což ale v tomto případě nebylo zapotřebí. Podrobné statistické vyhodnocení nebylo záměrem Autora.

3.1 Demografické údaje

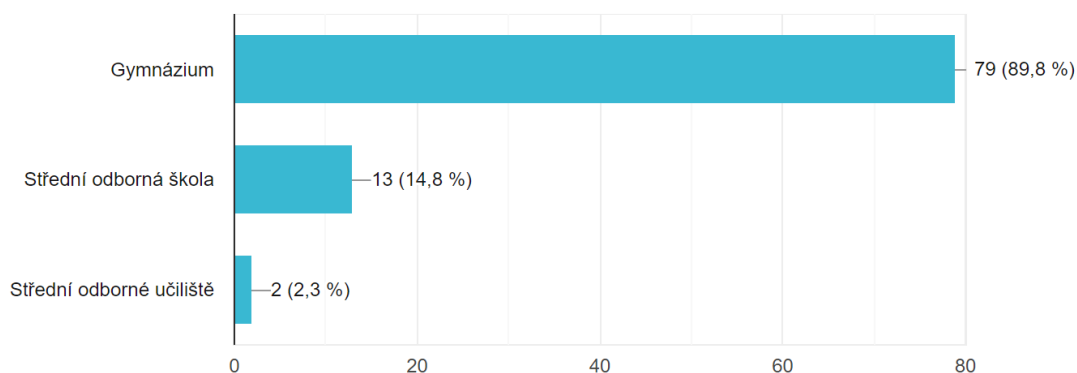
Otázky v této části dotazníku:

- Samosprávný kraj školy – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Zřizovatel školy – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Typ školy – uzavřená otázka – výběr více odpovědí z nabízených možností
- Zaměření školy – uzavřená otázka – výběr více odpovědí z nabízených možností
- Výuka programování – uzavřená otázka – výběr jedné odpovědi z nabízených možností

Dotazník vyplnilo 88 respondentů, jednalo se nejčastěji o učitele z hl. města Prahy (přibližně 21 % - 16 respondentů), poté v pořadí Jihomoravský kraj (17 % - 15 respondentů) a Moravskoslezský kraj (přibližně 11 % - 10 respondentů). Odpovědi z ostatních krajů se pohybovali v jednotkách procent.

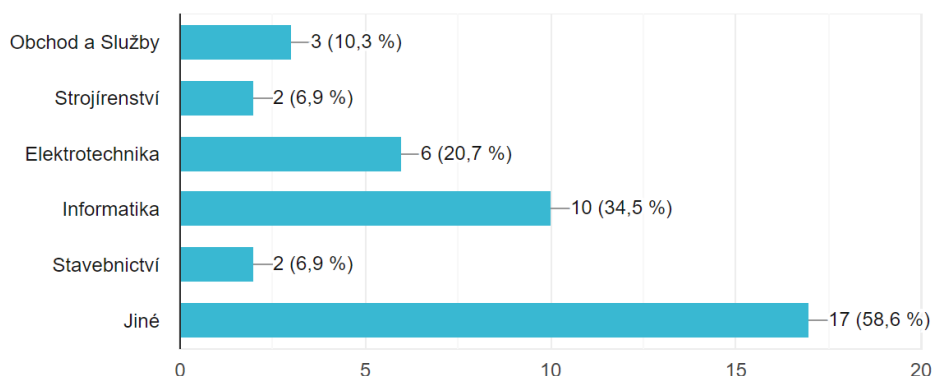
Nejčastějším zřizovatelem škol byl stát, tedy kraj, město nebo obec (přibližně 82 % - 72 respondentů), soukromý zřizovatel a církevní školy byly v menšině v řádech jednotek respondentů.

Téměř 90 % učitelů (79 respondentů) uvedlo, že vyučují na gymnáziu. Následné výsledky dotazníku jsou tedy primárně zaměřeny pro gymnázia (je však třeba poznamenat, že počet respondentů není na zobecnění dostačující, jelikož se nejedná o dostatečně velký statistický vzorek). Odpovídajících učitelů z průmyslových škol bylo 13 a učitelů z odborných učilišť pouze 2.



Obrázek 15 dotazník – zastoupení typů středních škol

Nejčastější odpověď, kterou respondenti zvolili v otázce týkající se zaměření školy byla možnost jiné, což koresponduje s velkým zastoupením gymnázií v rámci středních škol. Další nejpočetnější odpovědi byly informatika a elektrotechnika.



Obrázek 16 dotazník – zaměření středních škol

U této otázky bylo možné vybrat více typů škol, jelikož ve většině případů jsou například průmyslové školy zaměřeny na dva a více oborů.

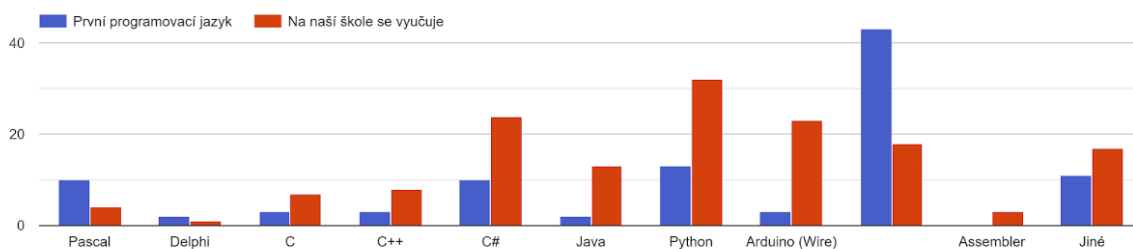
Na 90 % škol se programování vyučuje, ať už se bavíme o povinném předmětu v rámci oboru (34 respondentů – přibližně 38 %) a nepovinných seminářů nebo kroužků (46 respondentů – přibližně 52 %). Pouze 8 učitelů odpovědělo na tuto otázku možností „ne“ (v takovém případě byly rovnou přesměrováni na konec dotazníku).

3.2 Programování

Otázky v této části dotazníku:

- Programovací jazyky – uzavřená otázka – výběr více odpovědí z nabízených možností
- Arduino povědomí – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Arduino osobní zkušenost – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Arduino ve výuce – uzavřená otázka – výběr jedné odpovědi z nabízených možností

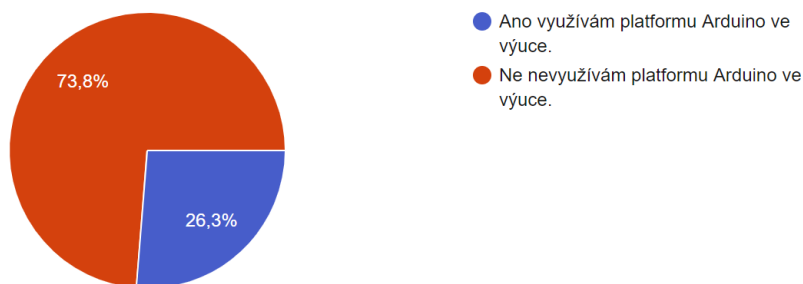
První otázka části programování, se zabírala programovacími jazyky. Modrý sloupec znázorňuje programovací jazyky, které jsou využity jako nástroj pro první vhled do světa programování. Červený sloupec pak označuje ty jazyky, které jsou na škole respondenta vyučovány (možnost, která není na obrázku níže popsána: „Scratch nebo jiný grafický programovací jazyk“). V této otázce bylo možné vybrat více odpovědí.



Obrázek 17 dotazník – vyučované programovací jazyky

Pokud se zaměříme pouze na modrou část, je možnost grafické programování, suverénně největší. Což není překvapivé, jelikož grafické programovací jazyky jsou nejlepší možnost, jak představit žákům programování. Stále je patrný vliv pascalu, dále pak pythonu a c#.

Další otázky se již týkaly pouze platformy Arduino. O projektu Arduino má povědomí přibližně 88 % učitelů (71 respondentů). Pouze 43 respondentů odpovědělo kladně na otázku, zda mají osobní zkušenost. Nejzajímavější je informace o tom, že pouze 21 učitelů využívá Arduino k výuce.



Obrázek 18 dotazník – využití platformy Arduino ve výuce

3.3 Arduino ve výuce

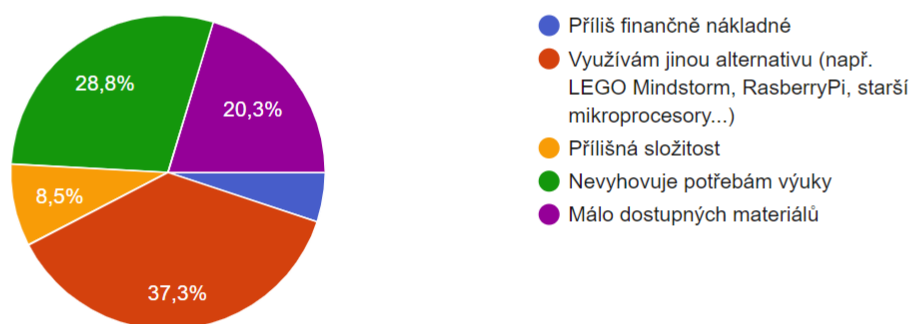
V závislosti na odpovědi z poslední otázky z předchozího bloku se dále dotazník dělil na dvě sekce.

3.3.1 Nevyužití

Otázky v této části dotazníku:

- Arduino ve výuce – uzavřená otázka – výběr jedné odpovědi z nabízených možností

V případě, že respondent odpověděl negativně, tedy že platformu Arduino ve výuce nevyužívá, následovala otázka zjišťující důvody k tomuto rozhodnutí.



Obrázek 19 dotazník – důvody nevyužití platformy Arduino ve výuce

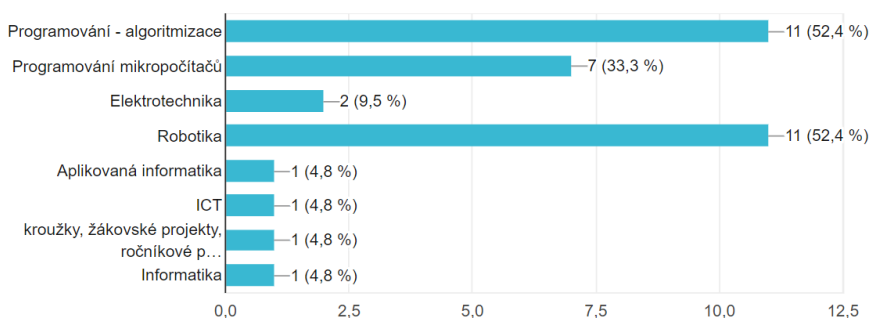
Nejpočetněji respondenti zvolili odpověď týkající se využití jiné alternativy. Druhá byla odpověď nevyhovuje potřebám výuky. Jen zanedbatelný počet respondentů zvolil možnost finanční nákladnosti.

3.3.2 Využití

Otázky v této části dotazníku:

- Arduino konkrétní předměty – polouzavřená otázka – výběr více odpovědí z nabízených možností, možnost vlastní odpovědi
- Arduino přednosti – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Arduino zápis kódu – uzavřená otázka – výběr jedné odpovědi z nabízených možností
- Arduino simulátory – uzavřená otázka – výběr jedné odpovědi z nabízených možností

Pokud respondent zvolil možnost, že ve výuce Arduino využívá následovaly tři otázky vztahující se ke konkrétním informacím.



Obrázek 20 dotazník – konkrétní příklady využití ve výuce

Nejčastější odpověď využití ve výuce byla robotika a překvapivě i obecná algoritmicizace. Druhou nejčastější odpovědí pak programování mikropočítačů. V této otázce bylo možné vybrat více odpovědí.

Jako největší přednost označili učitelé cenovou dostupnost. Nejčastější vývojové prostředí využívané ve výuce je Arduino IDE. Pouze jeden respondent uvedl že ve výuce simulátory využívá. Třináct respondentů pak o simulátorech někdy slyšeli.

3.4 Shrnutí

Výsledky statistického šetření do jisté míry potvrdili základní předpoklady. Platforma Arduino je obecně známá, a relativně velké procento učitelů s ní má osobní zkušenost. Na druhou stranu, ale není v praktické výuce využívána, což vyvrací prvotní premisu.

Důvody nevyužití jsou nejčastěji nahrazení alternativou nikoli cenová nedostupnost (jak bylo předpokládáno). Často se však v dotazníku v sekci, kde mohl respondent uvést vlastní poznatky a názory nejen k dotazníku objevovala jiná možnost. Nedostatečná časová dotace.

Jelikož se tyto údaje vztahují primárně ke gymnáziím, byly výukové materiály primárně koncipované jako alternativa ke klasickému programování právě na těchto typech škol, čemuž odpovídá i navrhovaný časový rámec celého kurzu a jednotlivých hodin.

Je však nutné znovu zopakovat, že počet respondentů není dostatečný ke zobecnování statistických výsledků.

4 VÝUKA PROGRAMOVÁNÍ

Programování ve své podstatě není pouze zápis algoritmu pro řešení konkrétního problému v konkrétním jazyce. Programování by mělo studenty naučit přemýšlet specifickým způsobem. Poznatky získané při řešení algoritmických úloh lze snadno převést do běžného, reálného života.

Výuka programování, nebo algoritmizace obecně, má přesah i do ostatních předmětů, spíše tedy technického nebo přírodovědného rázu. Ať je to matematika nebo fyzika. Ale můžeme se bavit i o vývoji funkčních aplikací pro vlastní potřebu, nebo psaní funkcí a maker pro zpracování dat, například v programu MS Excel.

4.1 Programovací paradigmatata

Programovací paradigmatata (návrhové vzory) jsou způsoby přístupu ke konstrukci logiky algoritmu. Historicky vzniklo několik návrhových vzorů, v závislosti na technologickém pokroku (rozsahu programu) nebo na požadavcích pro konkrétní oblast výpočetní techniky.

V následující kapitole jsou popsány a vysvětleny principy dvou nejznámějších návrhových vzorů.

4.1.1 Procedurální paradigma

Procedurální paradigma, někdy označované jako klasické nebo imperativní. Program je zápsán jako sekvence příkazů, která je v daném pořadí vykonávána. Jako stavební kámen tohoto přístupu je přiřazovací příkaz. Tento návrhový vzor je možné dělit na dvě podskupiny, a to strukturované a nestrukturované procedurální programování. [17; 18]

Nestrukturované programování je postaveno na příkazu GOTO (volně přeloženo jako: „skoč na řádek“). Tento příkaz po vykonání „skočí“ na místo v programu, označené návěštím, nebo číslem řádku. Tento styl velmi ztěžuje samotný zápis a ladění celého programu. Zápis algoritmu tímto způsobem se svou strukturou velmi podobá zápisu v Assembleru. [18]

Strukturované procedurální programování příkaz GOTO striktně zavrhuje. Strukturované programování nahrazuje tento příkaz pomocí podmíněných cyklů nebo jinou vhodnou programovací strukturou. [18]

4.1.2 Objektově orientované paradigma

Moderní způsob návrhu složitých programů je postaven na práci s objekty. Pod pojmem objekt si můžeme představit předměty z reálného světa. Člověk, zvíře, smlouva nebo krabice. Tyto objekty mají určité vlastnosti, například jméno, druh nebo číslo smlouvy. Tyto objekty obsahují procedury, které zajišťují samotnou funkcionalitu objektu. Program pak pracuje s interakcí mezi objekty, které si mezi sebou vyměňují informace. [17; 18]

4.2 Výběr jazyka

Výběr programovacího jazyka je počátečním bodem samotné výuky programování. Tato kapitola se věnuje několika obecným kritériím při výběru konkrétního řešení. [19; 20; 21]

4.2.1 Cílová skupina

Jedním z nejzákladnějších hledisek je cílová skupina. V tomto případě je možné cílovou skupinu rozdělit nejen podle věku, ale i podle studijního oboru, potažmo typu střední školy.

U mladších žáků na základní škole se převážně jedná o programování v logické rovině. Nejde tedy primárně o zápis algoritmu, ale spíše o samotné algoritmické myšlení při řešení konkrétních problémů. K tomuto účelu se nejvíce hodí grafické programovací jazyky jako LOGO Imagine, Scratch a další. [22]

V závislosti na studijním oboru (resp. typu školy) je vhodné zvolit jazyk, který může mít přesah do dalších předmětů, například databázi, návrhu webových nebo mobilních aplikací a podobně.

4.2.2 Aktuálnost

Jelikož nám jde o motivování studentů, je potřeba vybírat z „aktuální nabídky“ programovacích jazyků. Nelze očekávat velký zájem studentů při vyučování zastaralého programovacího jazyka, který se sami vyučující učili na střední škole před 30 lety. Studenti nejsou motivováni, vyučovat se něčemu, co už dopředu ví, že v reálné praxi nevyužijí.

S tímto bodem je úzce spojeno i programovací paradigma. V dnešní době je v praxi nejčastěji využívaný objektově orientovaný přístup, a velmi pravděpodobně se to v nejbližší době nezmění. Na druhou stranu je v tomto případě nutné vysvětlit další struktury a zákonitosti navíc, než pokud bychom například vyučovali starším procedurálním způsobem, kde se snáze zaměříme na samotnou algoritmizaci.

4.2.3 Dostupnost

Hodinová dotace často neposkytne dostatečný prostor na procvičení některých pokročilejších technik. Studenti by tedy měli mít možnost, si danou problematiku nastudovat a procvičit i mimo školní lavice.

Pokud budeme například pracovat v programovacím jazyku Swift, který je možné natively zprovoznit pouze na výrobcích společnosti Apple a ve verzi pro Linux [23], velmi tím omezujeme studenty, kteří k těmto platformám nemají snadný přístup. Toto se dá částečně obejít pomocí online webových kompilátorů, ale ani to není nejšťastnější řešení.

4.2.4 Syntaxe

Programovací jazyk vhodný pro výuku by neměl studenty zatěžovat složitým zápisem samotného kódu. Syntakticky by měl být co možná nejjednodušší. Pro tyto účely jsou nejvíce vhodné grafické programovací jazyky, které ve své podstatě téměř neumožňují syntaktickou chybu.

Na druhou stranu je v těchto programovacích jazycích poměrně komplikované vytvářet složitější programy. Takovéto programování navíc není příliš využitelné v praxi. Proto je vždy nutné najít vhodný kompromis mezi komplexností a náročností zápisu.

4.2.5 Literatura

Posledním bodem je dostupnost naučné literatury. V dnešní informační době je to pravděpodobně nejméně kritický bod, jelikož literatura (manuály, návody nebo například videotutoriály) je na internetu dostupná v dostatečně velké míře ke všem známým programovacím jazykům. Ale i na toto by měl vyučující myslet při výběru programovacího jazyka.

4.3 Platforma Arduino a výuka

Projekt Arduino vznikl v roce 2012, a každým rokem je stále více a více populárnější v poli mikropočítačové techniky. Dá se tedy tvrdit, že se jedná o moderní a aktuální platformu. Studenty dokáže zaujmout i přesahem do reálného světa. Není to jen o abstraktním kódu nul a jedniček. Jsou to tlačítka, LED a senzory.

Cena originálních desek je větší ve srovnání s klony jiných výrobců, srovnatelné funkčnosti (jak bylo popsáno v první kapitole). Za cenu originální desky Arduino Uno je ale možné z čínských velkoobchodů (Aliexpress, Alibaba, ...) objednat kompletní sadu, včetně

široké škály příslušenství. Bezplatnou možností je využití simulátorů, které jsou na internetu k dispozici. Základní vývojové prostředí je dostupné na všechny největší platformy – Windows, IOS a Linux. Studenti tedy mají relativně snadnou možnost pracovat i mimo běžnou školní výuku.

Výhoda platformy Arduino je možnost uzpůsobit zápis kódu konkrétním potřebám výuky, školy, učitele. Je možné programovat několika různými způsoby zápisu kódu. Programování v jazyce assembler, programování pomocí grafických editorů nebo zápis v jazyku C/C++. Poslední možnost umožňuje programovat klasickým, procedurálním způsobem, nebo vytvářet i objektově orientované programy.

Projekt Arduino si za dobu své existence získal mnoho příznivců a díky početné uživatelské základně je dostupných velká spousta návodů a rad, jak pracovat nejen se samotnou deskou, ale i s jednotlivými externími komponentami. Dostupnost literatury tedy v tomto případě není žádný zásadní problém. Například:

- Originální jazyková příručka spravovaná přímo společností Arduino. Ukázky jednotlivých příkazů jsou demonstrovány na praktických ukázkách včetně zapojení. Dostupné na: <https://www.arduino.cc/reference/en/>. Nevýhodou je nutnost znalosti anglického jazyka.
- Projekt portálu Arduino.cz – Průvodce světem Arduina. Komplexní pohled do základních konstrukcí včetně zajímavých ukázkových příkladů. Knihu je možné zakoupit, případně stáhnout 1. vydání v elektronické podobě, které autoři poskytli zdarma. Dostupné z: <https://arduino.cz/>. Kniha je kompletně v českém jazyce.

Projekt Arduino svou univerzálností poskytuje komplexní zázemí pro výuku širokého spektra různých předmětů. Různé způsoby zápisu a způsoby návrhu programu umožňují využití pro různé věkové kategorie. Nabízí zajímavější alternativu klasickému programování, díky propojení s reálným světem.

II. PRAKTICKÁ ČÁST

5 RÁMCOVÉ VZDĚLÁVACÍ PROGRAMY

V této kapitole je vysvětlen pojem RVP, vztah s ostatními školními dokumenty a jeho implementace na školách. Dále jsou pak rozebrány RVP a ŠVP gymnázia a průmyslové střední školy se zaměřením na předměty ve kterých se vyučuje programování. Tyto ŠVP jsou poté porovnány a určeny klíčové kapitoly, podle kterých byly stanoveny témata k vypracování.

5.1 RVP obecně

„Rámcové vzdělávací programy (RVP) tvoří obecně závazný rámec pro tvorbu školních vzdělávacích programů škol všech oborů vzdělání v předškolním, základním, základním uměleckém, jazykovém a středním vzdělávání. Do vzdělávání v České republice byly zavedeny zákonem č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon). Tento zákon byl novelizován v roce 2015 pod č. 82/2015.“ [24]

Rámcové vzdělávací programy jsou vytvořeny v souladu s Národním programem rozvoje vzdělávání v ČR (tzv. Bílé knihy), který definuje základní požadavky vzdělávání. Rámcové vzdělávací programy jsou zpracovány pro různé typy škol a oborů. V základu určují stejné základní požadavky. [25]

- Stanovuje základní vzdělávací úroveň pro absolventy
- Určuje úroveň klíčových kompetencí absolventů na konci vzdělávání ve vztahu k předmětům
 - kompetenci k učení
 - kompetenci k řešení problémů
 - kompetenci komunikativní
 - kompetenci sociální a personální
 - kompetenci občanskou
 - kompetenci k podnikavosti
- Stanovuje závazný vzdělávací obsah
 - Vzdělávací oblasti (předměty)
 - Očekávané výstupy a učivo
- Doporučuje způsob zařazení průřezových témat

Na základě RVP poté každá jednotlivá škola sestavuje vlastní ŠVP (školní vzdělávací program). ŠVP je následně realizován ve formě tematického plánu pro jednotlivé předměty.

V tematickém plánu jsou rozvrženy jednotlivá témata do časového harmonogramu školního roku.

5.2 RVP Gymnázia

Další část se zaměřuje na RVP určeného pro čtyřleté gymnaziální studium. Konkrétně pak předmět informatika. Dále je na konkrétním příkladu prezentován školní vzdělávací plán sestavený na základě RVP.

5.2.1 RVP – Informatika a informační a komunikační technologie

Pro předmět informatika a informační a komunikační technologie je v RVP vyhrazena minimální dotace čtyř hodin za celé studium. V případě gymnázia je studentům umožněno se do větší hloubky věnovat programování v rámci volitelných seminářů na konci roku. Tyto semináře mají studenty připravit na maturitní zkoušku právě z tohoto předmětu v profilové části zkoušky.

Úvodní část definuje základní úroveň práce s informacemi za použití digitální techniky. Implementaci a etický způsob využití v informační době.

5.2.1.1 Cílové zaměření vzdělávací oblasti

Cíle definované v rámci tohoto předmětu jsou následující (v tomto výčtu byly uvedeny pouze cíle, související s výukou programování/algoritmizace). [25]

- porozumění zásadám ovládání a věcným souvislostem jednotlivých skupin aplikačního programového vybavení a k vhodnému uplatňování jejich nástrojů, metod a vazeb k efektivnímu řešení úloh
- uplatňování algoritmického způsobu myšlení při řešení problémových úloh
- porozumění základním pojmům a metodám informatiky jako vědního oboru a k jeho uplatnění v ostatních vědních oborech a profesích

5.2.1.2 Vzdělávací obsah

V kapitole vzdělávací obsah jsou cílová zaměření podrobněji rozepsaná v kontextu očekávaných výstupů a samotného učiva na několik bloků. V případě programování (algoritmizace), které je zmíněno v bloku *Zpracování a prezentace informací* se jedná pouze o strohý popis probírané látky: „*algoritmizace úloh – algoritmus, zápis algoritmu, úvod do*

programování“. Očekávaný výstup související s programováním pak: „aplikuje algoritmický přístup k řešení problémů“ [25]

Samotné RVP nestanovuje konkrétní náplň hodin, ani podrobnější informace o samotné výuce. Toto je stanoveno v Školním tematickém plánu, který obvykle sestavuje předmětová komise s následným souhlasem ředitele školy. Dále je vhodné si povšimnout, že samotné RVP neurčuje ani programovací jazyk, nebo návrhový vzor.

5.2.2 ŠVP – Gymnázium Uherské hradiště – Vyšší stupeň vzdělávání

V této lekci bylo zkoumáno ŠVP Gymnázia Uherské Hradiště, Labyrint Světa II (pro vyšší gymnázium). Tato konkrétní škola byla zvolena z důvodu osobní zkušenosti se studiem autora.

Povinná výuka informatiky zde probíhá v prvních dvou ročnících (resp. kvinta a sexta), vždy dvě hodiny týdně.

Tabulka 2 ŠVP GUH – Informační technologie [26]

Školní výstupy	Učivo
Aplikuje algoritmický přístup k řešení problémů	Algoritmy, jeho vlastnosti a způsoby Zápisu; Programovací jazyky

Povinná část algoritmů je probírána pouze okrajově, z důvodu nedostatečné časové dotace. Jedná se o obecný vhled do algoritmizace a programovacích jazyků.

Učivo volitelného předmětu programování je o poznání obsáhlejší a zahrnuje konkrétní témata. V tabulce níže je uveden úvodní seminář pro třetí ročníky. V případě čtvrtého ročníku, je seminář orientovaný na grafické aplikace v programovacím jazyku C# se zaměřením na objektově orientované programování. [27]

Tabulka 3 ŠVP GUH – Programování pro 3. ročník

Školní výstupy	Učivo
Student si utřídí elementární znalosti, seznámí se základními pojmy oboru.	Pojem informatika, algoritmus
Student řeší jednoduché praktické slovní úlohy s cílem přiblížit obor pro řešení problémů návrhem algoritmu	Slovní algoritmy, příklady
Student se seznámí se základními vlastnostmi algoritmu s tím, že je upozorněn na rozdíl mezi běžným algoritmem – návodem, a naopak matematickým pojmem algoritmus	Algoritmus a jeho zápis

Základní činnost algoritmického zpracování dat je realizace cyklu, postupně jsou na jednoduchých příkladech přibližovány jednotlivé zásady řešení problémů algoritmického charakteru jednoduchým cyklem	Jednoduché cykly
Student se seznámí se základními cíli a metodami správného postupu při vytváření algoritmů, je na praktických příkladech seznámen s uznávanou metodologií pro psaní algoritmů	Strukturované programování
Student řeší náročnější úlohy praktické povahy vyžadující vnořený cyklus – množinové operace, maticové operace a třídící algoritmy	Vnořené cykly
Student zvládne nástroj pro zapisování algoritmů do počítače	Zápis v programovacím jazyku – TURBO Pascal
Student pozná základní nástroje na porovnávání kvality algoritmů, umí zhodnotit jejich složitost a tím i jejich efektivitu	Složitost algoritmů
Nejefektivnější a nejmodernější nástroje pro zápis algoritmů jsou představeny v jednoduchých i složitějších variantách	Rekurze, backtracking

V tomto předmětu je již deklarován i konkrétní programovací jazyk. V tomto případě se jedná o jazyk TURBO Pascal, který je již velmi zastaralý a v reálné praxi příliš nepoužívaný.

5.3 RVP průmyslová škola

Jako další ukázkový typ školy byla zvolena Průmyslová škola, obor Technické lyceum (78-42-M/01). Konkrétní ŠVP je poté vysvětleno na příkladu Střední průmyslové školy Zlín.

5.3.1 RVP

Technické lyceum je obecná technická příprava. Tento obor byl vybrán, jelikož se jedná o obecnou přípravu, stejně jako gymnázium, v tomto případě však technického zaměření.

5.3.1.1 Vzdělávací obsah

Část zabývající se programováním je uvedena v předmětu vzdělávání v informačních a komunikačních technologiích.

Školní výstupy	Učivo
ovládá principy algoritmizace úloh a je schopen sestavit algoritmus řešení konkrétní úlohy (dekompozice úlohy na	algoritmizace programování

<p>jednotlivé elementárnější činnosti za použití přiměřené míry abstrakce)</p> <p>vytvoří a odladí jednoduchý program v některém vývojovém prostředí</p> <p>ovládá základy objektově orientovaného programování.</p>	
--	--

Stejně jako v předchozím případě, ani zde nenajdeme podrobnější informace o využitém programovacím jazyku a programovacím paradigmatu.

Pro výuku informačních technologií je v RVP stanovena časová dotace 10 hodin týdně. Tedy za běžnou výuku 56 týdnů, jedná se o 560 hodin za celý školní rok.

5.3.2 ŠVP – Střední průmyslová škola Zlín Technické lyceum

Obor technické lyceum obsahuje samostatný předmět programování. Tento předmět je do výuky zařazen ve druhém ročníku a je pro něj vyhrazena časová dotace tří hodin týdně. (Kvůli obsáhlosti bylo uvedené ŠVP zjednodušeno).

Tabulka 4 ŠVP SPSZL – Technické lyceum – programování pro 2. ročník

Školní výstupy	Učivo
<p>charakterizuje algoritmickou úlohu</p> <p>sestavuje výrazy pro transformaci dat</p> <p>vysvětlí postup při algoritmizaci úloh</p> <p>sestavuje algoritmy konkrétních úloh</p> <p>znázorňuje správně sestavené algoritmy v grafickém jazyku (např. vývojový diagram)</p> <p>používá trasovací tabulku k dokazování správnosti algoritmu</p>	<p>algoritmus, vlastnosti, formy zápisu</p> <p>algoritmická úloha, proměnné</p> <p>transformace dat, operátory, výrazy, příkaz dosazení</p> <p>typy rozkladů, základní algoritmické konstrukce</p> <p>použití posloupnosti, alternativ, multivětvení a cyklů při algoritmizaci úloh</p> <p>kontrola správnosti algoritmu trasovací tabulkou</p>
<p>vysvětlí postup vývoje programu</p> <p>rozumí pojmu syntax programovacího jazyka</p> <p>objasní základní strukturované příkazy (sekvenci, alternativu, multivětvení, cykly)</p> <p>vysvětlí pojmy identifikátor a datový typ</p> <p>dokáže charakterizovat jednoduché datové typy</p> <p>sestaví a odladí jednoduchý program (bez použití podprogramů)</p>	<p>vývoj programu, struktura zdrojového textu</p> <p>lexikální prvky programovacího jazyka, syntaktická pravidla</p> <p>zápis základních algoritmických konstrukcí v programovacím jazyku</p> <p>zápis sekvencí vývojových diagramů do programovacího jazyka</p> <p>deklarace konstant, proměnných</p> <p>procedury vstupu a výstupu dat</p> <p>jednoduché datové typy, přípustné hodnoty a operace</p> <p>tvorba programů s jednoduchými typy</p>
<p>popíše návrh programu metodou shora dolů</p> <p>vysvětlí rozdíl mezi formálním a skutečným parametrem, lokální a globální proměnnou</p>	<p>formát deklarace podprogramů</p> <p>lokální a globální proměnné</p>

objasní mechanismus předávání parametrů používá podprogramy při programování jednodušších úloh	komunikace (formální a skutečné parametry) aktivace podprogramu návrh programu metodou shora dolů tvorba programů s využitím podprogramů
používá standardní podprogramy pro práci s textem	deklarace, povolené operace a standardní podprogramy tvorba podprogramů nad typem řetězec
charakterizuje strukturované datové typy (pole a záznam) definuje selektor prvku pole a záznamu vysvětlí operaci hledání význačného prvku v poli využívá při tvorbě algoritmů strukturované datové typy (pole a záznam)	datový typ pole, deklarace, povolené operace generování prvků pole, výpis prvků pole, hledání max. a min. prvku pole tvorba podprogramů nad typem pole struktury (záznamy), deklarace, povolené operace tvorba podprogramů nad typem záznam tvorba podprogramů nad datovou strukturou vzniklou kombinací pole a záznamu
objasní princip jednotlivých metod řazení prvků pole vysvětlí princip jednotlivých metod vyhledávání	základní metody řazení: BubbleSort, SelectSort, InsertSort vyhledání prvku v neseřazeném poli vyhledání prvku v seřazeném poli tvorba podprogramů jednotlivých metod řazení a vyhledávání
popíše princip objektového programování vysvětlí vztah mezi třídou a objektem	základní myšlenka OOP zapouzdření metod a vlastností do objektů třída a objekt

5.4 Srovnání

Témata výukových materiálů pro tuto práci byla vybrána na základě srovnání předchozích ŠVP. Vypracované témata jsou následující:

- Pojem informatika, algoritmus – typy a vlastnosti
- Slovní algoritmy – příklady
- Algoritmus a jeho zápis – jazyk vývojových diagramů, zápis v jazyku Arduino
- Proměnné a datové typy – deklarace, přiřazení, práce s proměnnými
- Konstanty a datové typy – deklarace, přiřazení, práce s konstantami
- Větvení – podmínky, složená podmínka
- Jednoduché cykly – typy, využití, zápis
- Pole – deklarace, práce s polem, indexy, hledání maxima a minima
- Funkce – deklarace, volání,

Pro práci s platformou Arduino bylo nutné přidat několik specifických témat, které nevycházejí přímo z ŠVP, jako například vstupy a výstupy, PWM a další.

6 PŘÍPRAVA HODINY

Hlavní částí diplomové práce bylo zpracovat přípravy hodiny pro učitele a pracovní listy pro žáky. Přípravy byly následně doplněny prezentacemi pro podporu výkladu.

Myšlenkou autora bylo využití platformy Arduino pro výuku klasické algoritmizace, která by tímto dostala přesah i do reálného světa a nejednalo by se pouze o abstraktní programování. Za použití vhodného technického vybavení (výukové shieldy nebo výukové moduly) by bylo možné vynechat problémy se zapojením a minimalizovat nutné elektrotechnické znalosti, tak aby bylo možné soustředit se pouze na programování.

Samozřejmě je platforma Arduino ze své podstaty mikropočítače určena hlavně do předmětů, zabývajících se mikropočítači. Stejně tak i elektrotechnicky zaměřených předmětů, kvůli využití elektrotechnických součástek. I pro tyto předměty by měly být materiály vhodným základem výuky.

6.1 Materiály pro učitele

V této kapitole jsou popsány jednotlivé části materiálů pro učitele. Ukázkový materiál je přiložen jako Příloha II.

6.1.1 Didaktická část

Úvodní část materiálů pro učitele obsahují základní didaktické požadavky na přípravu jednotlivých vyučovacích hodin.

6.1.1.1 *Téma vyučovací hodiny*

V této části je uvedeno téma vyučovací hodiny. Tedy konkrétní učivo, důležité pojmy a vztahy mezi nimi.

6.1.1.2 *Cíl hodiny*

Kognitivní cíle hodiny byly stanoveny pomocí Bloomovi taxonomie. Cíle se vždy vztahují k očekávaným žákovským výstupům a jsou vyjádřeny jako činnost žáka pomocí typických sloves. Výukové cíle se podle Bloomovi taxonomie dělí na několik kategorií. Uvedené rozložení je členěno podle úrovně poznání. [28; 29]

1. **Znalost (zapamatování)** – vybavení poznatků a jejich reprodukce, nikoli praktické využití (definovat, opakovat, pojmenovat, ...)

2. **Porozumění** – pochopení poznatků a jejich reálné využití (dokázat, uvést příklad, vysvětlit, vyjádřit vlastními slovy, ...)
3. **Aplikace** – využití poznatků na situacích pro žáka doposud neznámých a problémových (demonstrovat, aplikovat, diskutovat, řešit, uvést vztah mezi, ...)
4. **Analýza** – rozložení poznatků na části, s objasněním vzájemných vztahů mezi nimi. Rozlišení důležitějších částí a vztahů od těch méně důležitých (najít princip, analyzovat, rozlišit, specifikovat, ...)
5. **Syntéza** – skládání prvků a částí do nových celků (skládat, kombinovat, syntetizovat, klasifikovat, ...)
6. **Hodnotící posouzení** – posouzení metod, řešení, výtvorů, na základě konkrétního kritéria (obhájit, posoudit, uvést klady a zápory, ...)

Tyto kategorie jsou uspořádané v hierarchickém pořadí. Úvodní premisa předpokládala, že zvládnutí kategorie vyšší úrovně vyžaduje vždy zvládnutí kategorie předchozí. Tuto domněnku se však nepodařilo ověřit. [28]

Základní struktura formulovaného požadavku se skládá z cílové skupiny (studenti, žáci), označení podmínek, a označení kritéria pro hodnocení. [29; 30]

Cíle, které se vztahují ke zvládnutí elektrotechnických zapojení patří do kategorie psychomotorických. V případě afektivních cílů nebyly uvedeny konkrétní formulace. Tyto cíle je v případě výuky programování velmi složité realizovat. V návrhu výuky jsou tyto cíle částečně zahrnuty do způsobu řešení úkolů, u kterých se předpokládá prezentace studentských řešení a následná diskuse s vlastními názory na řešení.

6.1.1.3 Klíčové kompetence

„Klíčové kompetence představují soubor vědomostí, dovedností, schopností, postojů a hodnot, které jsou důležité pro osobní rozvoj jedince, jeho aktivní zapojení do společnosti a budoucí uplatnění v životě.“ [25]

Základní klíčové kompetence, na které je zaměřena navrhovaná výuka i výuka programování obecně jsou tyto:

- **Kompetence k učení** – práce s materiály, organizace vlastní práce, sebereflexe ve vztahu k učení
- **Kompetence k řešení problému** – samostatné úkoly, objasnění zadání, rozčlenění na dílčí části, navržení kroků k řešení

- **Kompetence komunikativní** – diskuze nad řešením, prezentace vlastních řešení

Podrobnější popis implementace jednotlivých klíčových kompetencí v rámci výuky je uvedena v RVP pro každý předmět.

6.1.1.4 *Organizační formy výuky*

V této kapitole jsou popsány navrhované organizační metody výuky využité ve vypracovaných výukových materiálech. [28]

- **Hromadná a frontální výuka** – Frontální výuka je stěžejním základem celé navrhované výuky.
- **Skupinová výuka** – při vypracování úkolů z hodiny je možné rozdělit studenty na menší skupiny nebo dvojice, které budou pracovat na úkolech společně. Vyučující může operativně upravit náročnost úkolu, tak aby odpovídal zkušenostem a schopnostem členů skupiny.
- **Projektová výuka** – tento typ je využit v rámci závěrečného projektu celého předmětu.

6.1.1.5 *Metody výuky*

Metoda je do jisté míry způsob dosažení stanovených výukových cílů. V této části materiálů jsou popsány výukové metody, navrhované pro výukovou hodinu. Členění metod odpovídá rozložení navrženému Josefem Maňákem podle pramene poznání a typu poznatku. [28]

- **Metody slovní: monologické metody** – předávání nové teoretické látky žákům. Zahrnuje různé formy – vysvětlování, popis, výklad. Jelikož se jedná o nejméně záživný způsob předávání informací, byla snaha tuto metodu omezit na co nejmenší rozsah a větší část hodiny využít ostatní metody, více zaměřené na práci studentů.
- **Metody slovní: dialogické metody** – diskuze nad řešenými příklady a úkoly, hledání nejlepšího řešení, srovnávání řešení.
- **Metody slovní: práce s učebnicí, pracovním textem** – dohledávání potřebných informací v pracovním textu, při řešení zadaných úkolů.
- **Metody praktické: žákovské laborování** – práce na zadaných úkolech

6.1.1.6 *Materiální didaktické prostředky*

V této části jsou vždy uvedeny požadované materiální didaktické prostředky potřebné pro práci v hodinách. Klasifikace jednotlivých položek, podle Josefa Malacha. Nejčastěji použité didaktické prostředky, při navrhované výuce jsou následující. [28]

- **Učební pomůcky:** Textové pomůcky – pracovní materiály
- **Technické výukové prostředky:** Audiovizuální technika – dataprojektor
- **Speciální pomůcky:** Žákovské experimentální pomůcky – deska Arduino a základní komponenty (tato kategorie neodpovídá přesně povaze desky Arduino, ale z nabízených kategorií nejvíce vyhovovala)
- **Výukové prostory a jejich vybavení:** Počítačová učebna

6.1.2 *Výuka*

Samotná část výuka je většinou strukturována v tomto pořadí. Teoretický úvod – vysvětlení základních pojmů a principů. Část příklady – vypracované příklady jako praktická ukázka teoreticky probírané látky a část úkoly – samostatná práce pro žáky.

Toto rozvržení odpovídá klasickému deduktivnímu přístupu k vyučování. Zahájení, budování významu, procvičení. Poslední část ověření výukových cílů (případné hodnocení) je závislá na úvaze učitele a typu výuky. Navrhované řešení k tomuto využívá hodnocení zadaných úkolů z jednotlivých hodin. [29]

6.1.2.1 *Teorie*

Na začátku každé vyučovací hodiny jsou žáci seznámeni s tématem a probíranou látkou, na kterou se zaměřuje daná hodina. Další část se již zabývá teorií jako takovou. Vysvětlení nových pojmů nebo vztahů mezi jednotlivými pojmy.

6.1.2.2 *Příklady*

Na příkladech mají studenti možnost pochopit základní principy nově probrané látky. Tato část je určená z větší části pro učitele, který tímto může na konkrétní úloze demonstrovat postupy přímo v programovacím jazyku Arduino.

Každý příklad se skládá z konkrétního zapojení s komentovaným kódem. U většiny příkladů byly vypracovány i interaktivní simulace, tak aby studenti měli možnost tyto příklady pozorovat v reálném čase.

6.1.2.3 Úkoly

Jedná se úkoly, na kterých mají žáci aplikovat znalosti získané z dané hodiny. Úkoly byly konstruovány tak, aby žáci museli využít i znalosti z předchozích hodin. Tímto byla zacílena jedna ze zásad výuky, a to zásada postupnosti.

Tato část je v případě materiálů pro učitele strukturovaná stejně jako část příklady. Schéma zapojení a vypracovaný kód s poznámkami.

Před vypracováním samostatných úkolů je vhodné alespoň částečně nastínit logiku programu. Nejlépe však metodou brainstormingu, kdy se zapojí primárně studenti. Stejně tak je vhodné věnovat dostatečný čas diskusi nad konkrétním studentským řešením.

6.2 Materiály pro žáky

Materiály pro žáky jsou z velké části totožné s přípravou hodiny pro vyučující. Některé části z nich však byly odstraněny. Konkrétně se jedná o didaktickou část (výukové cíle, metody, ...), poznámky pro učitele a část úkolů. V poslední jmenované je v materiálech uvedeno pouze zadání jednotlivých úkolů, bez řešení a zapojení.

Materiály pro studenty mají sloužit jako ucelený přehled všech teoretických a praktických informací, tak aby v případě potřeby měli informace k dispozici v ucelené a strukturované formě.

7 ZPRACOVANÁ TÉMATA

Následující kapitola obsahuje seznam vypracovaných témat. Jednotlivé lekce byly svým tématem sestaveny z průřezu školních a rámcových vzdělávacích programů z předchozích kapitol. Konkrétní realizace pak záleží na typu školy a požadavcích na učební obor. Některé kapitoly bude nutné rozšířit nebo případně vynechat.

Celá látka byla rozdělena na 4 bloky. Úvod, Základní HW komponenty, Pokročilé HW komponenty a závěrečný projekt. Navíc byl vypracován blok o obecné algoritmizaci.

Tabulka 5 seznam zpracovaných témat a jejich předpokládané časové rozvržení

Název bloku	Název kapitoly	Počet vyučovacích hodin
Algoritmizace 10–12 vyučovacích hodin	Úvod	2
	Základní struktura a zápis	2
	Cyklus	2
	Pole	2
	Procvičování	2 (4)
Arduino – úvod 16–18 vyučovacích hodin	Úvod	2
	Zápis a struktura programu	2
	Vstupy a výstupy	2
	Proměnné	2
	Podmínky	2
	Cykly	2
	Pole	2
	Procvičování	2 (4)
Arduino – základní HW komponenty 12–14 vyučovacích hodin	Sedmi segmentový displej	2
	Ultrazvukový senzor, funkce	2
	Piezoreproduktor	2
	Servo, PWM	2
	Senzor teploty	2
	Procvičování	2 (4)
Arduino – Pokročilé HW komponenty 10–12 vyučovacích hodin	LCD displej, knihovny	2
	Tlačítková matice	2
	PIR senzor	2
	LED pásy	2
	Procvičování	2 (4)
Arduino – závěrečný projekt 14–16 vyučovacích hodin	Práce na projektu	8
	Zpracování dokumentace	4
	Prezentace	2 (4)

Časové rozvržení celé výuky bylo koncipováno, jako dvouhodinové úseky, tak aby bylo vždy teoretické části zvládnutelné během jednoho týdne. V případě hodin, které jsou primárně určeny k procvičování, byla celková časová dotace navýšena.

7.1 Algoritmizace

Tento blok poskytuje teoretický úvod do problematiky algoritmů a algoritmického myšlení. Primárně slouží jako sjednocovací blok učiva tak, aby vědomosti studentů byly přibližně na stejné úrovni. V případě, že se nejedná o první předmět, který se zabývá programováním nebo algoritmizací, lze celý blok vynechat. Případně lze využít pouze jako opakování.

Celá tato část se sestává ze čtyř základních témat. Poslední téma slouží jako opakování. Celkově se tedy jedná o blok délky deseti až dvanácti vyučovacích hodin v závislosti na schopnostech a dovednostech třídy.

V této kapitole není zahrnuta práce s vícerozměrným polem ani třídící algoritmy. Tyto pokročilejší vědomosti nejsou klíčové pro programování platformy Arduino, v konceptu navržené výuky.

7.1.1 Úvod

Úvodní dvouhodina, ve které je představen pojem Algoritmus na praktických ukázkách z běžného života. Definice z pohledu počítačové techniky, včetně jeho základních vlastností.

7.1.2 Základní struktura a zápis

V této dvouhodinové lekci jsou popsány způsoby zápisu algoritmu. Pseudokód, jazyk vývojových diagramů a programovací jazyk.

Největší část je věnována jazyku vývojových diagramů, pomocí kterých je vysvětlován celý blok algoritmizace. Jako příklady byly využity příklady z předchozí hodiny, které mají studenti za úkol přepsat do jazyka vývojových diagramů.

7.1.3 Cyklus

Základní náhled do problematiky cyklů, jejich vlastnosti a typy. Cyklus s podmínkou nahoře a dole. Vysvětlení pojmu řídicí proměnná včetně demonstrace na praktických příkladech.

7.1.4 Pole

Vysvětlení pojmu pole a základních operací s ním. Ukázky práce s celým polem pomocí cyklů a přístup na jednotlivé prvky pomocí indexů.

7.1.5 Procvičování

Tato kapitole se sestává pouze z příkladů na procvičení. Primárně se zaměřuje na příklady pokrývající problematiku pole a práci s ním. Celkem bylo v této lekci vypracováno 6 příkladů, pokrývajících všechny základní znalosti z předchozích hodin. Této kapitole je vhodné věnovat více času než navrhované 2 hodiny.

7.2 Arduino – Úvod

V tomto bloku jsou popsány základní pojmy při práci s platformou Arduino. Jedná se spíše o teoretický úvod, do základních příkazů a programových struktur. Stejně tak i specifických pojmů, které se týkají platformy Arduino, jako jsou například digitální a analogové vstupně/výstupní piny.

Zpracované úlohy pro žáky jsou většinou spíše teoretického rázu. Slouží převážně k procvičení probíraného tématu. Zvláště pak v první polovině. V lekci vstupy a výstupy byla popsána práce se základními vstupními a výstupními prvky. Tlačítka, LED a potenciometr. Od této lekce byly úkoly zaměřeny více prakticky.

Konstrukce jsou hodiny orientovány spíše na učitele, jelikož v teoretických pojmech je velmi složité navrhnout jinou alternativu. V dalších blocích byl výklad učitele omezen na nutné minimum.

Při vytváření materiálů bylo primárně čerpáno z [31; 32; 33] a z oficiální webové příručky Arduino [34]

7.2.1 Úvod

V této lekci je obecné zamyšlení nad automatickým řízením. Obecně byl ve zkrácené formě popsán princip mikropočítače. Další část byla věnovaná teoretickému úvodu k platformě Arduino. Historie, nejběžnější typy desek, komponenty desky a pojem klony. Samostatná práce studentů je zaměřena na seznámení s praktickými projekty, realizované na platformě Arduino.

7.2.2 Zápis kódu a struktura programu

Vysvětlení instalace Arduino IDE. Založení nového projektu, uložení a načtení. Základní práce s IDE. Registrace na webové aplikace Tinkercad. Založení nového projektu, popis pracovního prostředí.

Dále je popsána základní struktura programu Arduino. Funkce setup a loop. Zapojení a spuštění prvního programu „HelloWorld“ světa Arduino (Blikání LED na desce Arduino).

7.2.3 Vstupy a výstupy

V této hodině je popsán princip digitálního a analogového signálu, dále pak analogově digitální převod.

Dále jsou popsány všechny vstupně výstupní piny na desce Arduino UNO, včetně rezervovaných pinů jako je napájení, pinu externího zdroje napětí (Vin), pinu referenčního napětí (GND) a dalších.

V další části je vysvětlena základní práce s digitálními vstupně/výstupními piny, jejich inicializace, čtení a zápis hodnot. Tyto příkazy jsou využity při vysvětlení zapojení LED a jednoduchého tlačítka. Další část se zaměřuje na analogový vstup a výstup s využitím potenciometru.

7.2.4 Proměnné

V této hodině je vysvětlen pojem proměnná a její základní datové typy, se kterými je možné pracovat (integer, float, double, char, string a bool). Příkazy deklarace proměnné, přiřazení hodnoty do proměnné a definice proměnné. V další části je vysvětlen pojem konstanta, její deklarace a použití v programu.

7.2.5 Podmínky

Tato hodina je zaměřena na vysvětlení pojmu větvení. Objasnění pojmu podmínka. Základní programové struktury, příkazy if, if-else. Další část se věnuje relačním operátorům AND a OR a s nimi spojené složené podmínky.

7.2.6 Cykly

V této lekci je popsán pojem cyklus a řídicí proměnná. Vysvětleny jsou cykly while, do-while a for. Pro práci s cykly je popsán i princip řídicích příkazů continue a break.

7.2.7 Pole

Vysvětlení pojmu pole, deklarace a základní operace nad polem – inicializace, přiřazení hodnoty do prvku pole, přístup na konkrétní prvek za pomoci indexu. Práce s polem pomocí cyklu for.

7.3 Arduino – Základní HW prvky

V tomto bloku jsou postupně představeny jednoduché komponenty připojitelné k desce Arduino. Navíc je vysvětleno využití a návrh vlastních funkcí a pojem pulsně šířková modulace. Na konec tohoto bloku byly navrženy příklady k procvičení, které využívají všech nabytých vědomostí z tohoto bloku.

7.3.1 Sedmi segmentový displej

V této lekci je popsán princip sedmi segmentového displeje, jeho zapojení a využití. V další části je vysvětlena direktiva `define` a příkaz `switch..case`. Oba pojmy jsou využity v ukázkovém příkladu a samostatné práci.

7.3.2 Ultrazvukový měřič vzdálenosti, funkce

V první části je popsán princip ultrazvukového senzoru vzdálenosti, včetně odvození rovnice použité pro práci s ním. Dále je vysvětlen způsob zapojení a základní obsluhy.

Další část je věnována návrhu funkcí. Je popsán způsob deklarace a její základní části (typ návratové hodnoty, identifikátor a parametry). Praktické příklady jsou zaměřeny spíše v teoretické rovině. Prakticky si studenti procvičí návrh funkce v samostatném úkolu, ve kterém je zadáno vytvoření funkce pro měření vzdálenosti z ultrazvukového senzoru. Poslední úkol, je zaměřen na využití ultrazvukového měřiče vzdálenosti k měření rychlosti.

7.3.3 Servo, PWM

Podrobnější vysvětlení pojmu pulsně šířková modulace a s ní spojený příkaz `analogWrite`. Dále je vysvětlen princip serva a jeho konstrukce. Další část je zaměřená na ovládání serva, pomocí pulsně šířkové modulace a příkazu `analogWrite`.

7.3.4 Piezo bzučák

V této lekci je vysvětlen piezoelektrický jev a jeho využití při generování zvuku s využitím piezo bzučáku. Dále je popsán princip příkazu `tone` pro generování signálu o požadované frekvenci.

7.3.5 Teplotní senzor

V této lekci je vysvětleno využití a princip teplotního senzoru TMP36. Způsob zapojení a jeho obsluha.

7.4 Arduino – Pokročilé HW prvky

Tento tematický blok je zaměřen na pokročilejší hardwarové komponenty, čemuž odpovídají i pokročilejší příklady a úkoly, u kterých je potřeba pokročilejší algoritmické myšlení. V první lekci je vysvětlen pojem knihovna a její implementace a využití v projektu.

Opět byly navrženy některé pokročilejší příklady k řešení. Tyto příklady je možné vynechat a o tento čas prodloužit závěrečnou kapitolu – samostatný projekt.

7.4.1 LCD displej

Tato lekce je zaměřena na princip a zapojení jednoduchého LCD displeje. Pro práci s displejem byla využita knihovna LiquidCrystal. Na tomto příkladu je vysvětlen i způsoby a důvody přidání externích knihoven do projektu.

7.4.2 Tlačítková matice

V této lekci je popsán princip tlačítkové matice a její připojení k desce Arduino. Pro aktivní využití byly popsány základní příkazy knihovny Keypad.

7.4.3 PIR senzor

V této lekci je vysvětlen princip PIR senzoru a jeho využití v praxi. Následně je popsán způsob zapojení k desce Arduino a jeho ovládání.

7.4.4 LED pásky

Tato lekce je zaměřena na popis principu LED pásek NeoPixel, způsob připojení k desce Arduino a základní příkazy knihovny AdafruitNeoPixel.

7.5 Arduino – Závěrečný projekt

Závěr celého předmětu je věnovaný samostatné práci studentů, řízené individuálně učitelem. Jako závěr tohoto bloku je předpokládána i prezentace jednotlivých studentských projektů. Hodinová dotace se bude mírně lišit podle počtu studentů ve třídě.

Pro potřeby závěrečného projektu byly navrženy některé možné variace na závěrečný projekt. Zadání by si však měli studenti, po konzultaci s vyučujícím, navrhnout do jisté míry sami. Obtížnost, je poté korigována právě vyučujícím.

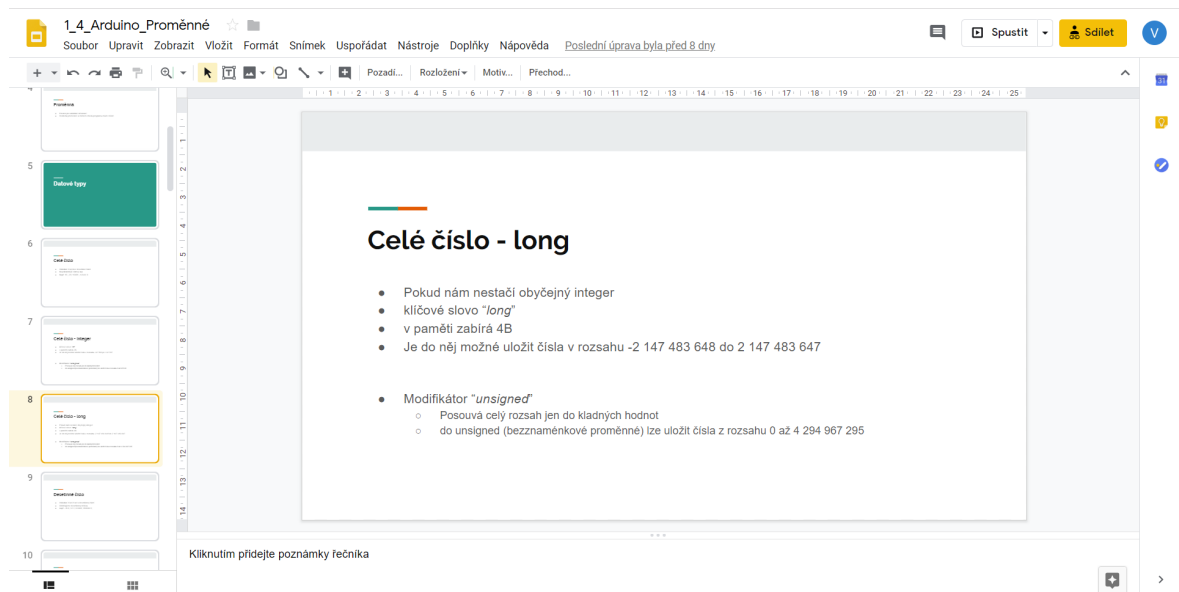
Závěrečný projekt může být založen na komponentech, které nebyly probrány na hodinách a studenti by tedy museli sami dostudovat zapojení a obsluhu. Toto je samozřejmě možné

uskutečnit primárně u nadanějších studentů. Množství probraných komponent je dostatečné pro vytvoření středně komplexního projektu i pro běžně nadané studenty.

Další částí závěrečného projektu je krátká dokumentace, ve které budou závěrečné projekty popsány do větších detailů, včetně komentovaného kódu. Tento bod, je možné vynechat, ovšem je to dobrý způsob, jak naučit studenty zpracovat výstup vlastní práce, potažmo technickou dokumentaci.

8 PREZENTACE GOOGLE

Ke všem kapitolám výuky byly vypracovány prezentace jakožto podpora výkladu učitele. K jejich vytvoření byla použita aplikace prezentace Google. Její použití bylo podmíněno dvěma důvody. Hlavním důvodem byla snadná implementace do webu, vytvořeného pomocí další aplikace firmy Google (popsáno v další kapitole). Dále pak snadná editace přímo z webové aplikace. Strukturou prezentace odpovídají přípravám výuky.



Obrázek 21 ukázka prostředí Prezentace Google

Webová aplikace Prezentace Google nabízí téměř stejnou funkčnost jako program powerpoint, který je součástí kancelářského balíku Microsoft office.

9 WEBY GOOGLE

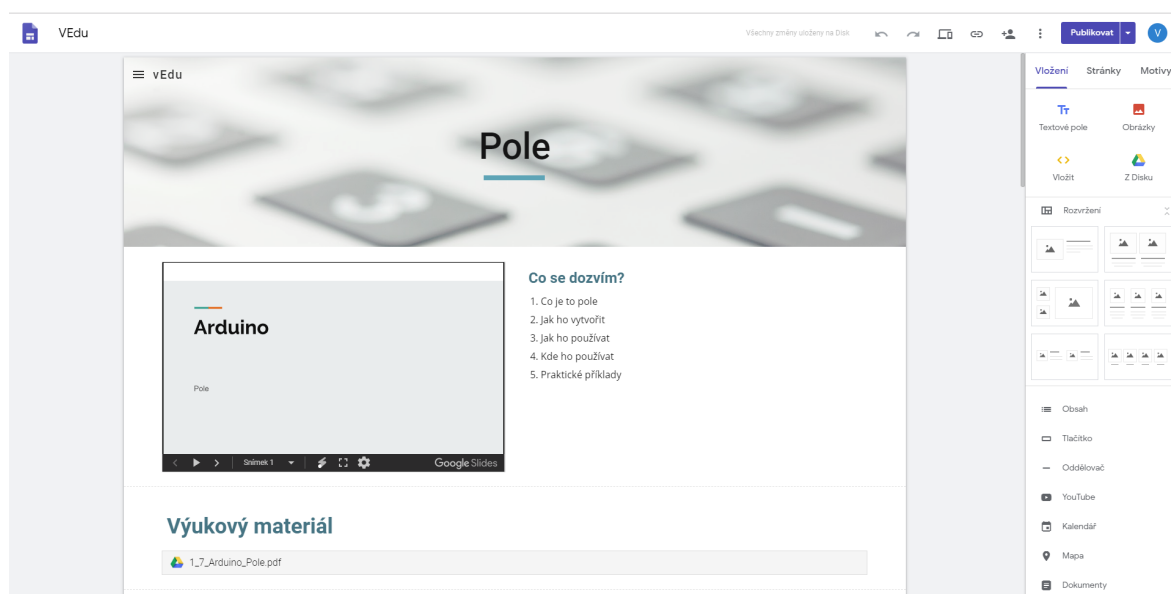
Dalším bodem této práce bylo vytvoření webových stránek, které by poskytovaly ucelenou platformu pro podporu výuky. Bylo zvažováno několik variant. Vlastní doména a některé domény zdarma. Pro jednoduchost bylo nakonec přistoupeno k vytvoření webové stránky pomocí služby Weby Google (dále uváděno anglickým názvem GoogleSites).

Aplikace GoogleSites je vázána na Google účet, z toho důvodu byl založen účet vrablik.education@gmail.com přímo pro účely této stránky a celé diplomové práce (a také další pedagogické činnosti autora). Firma Google nabízí ke každému účtu 15 GB cloudový prostor na aplikaci GoogleDrive zcela zdarma (Rozšíření je možné v případě placené verze GoogleDrive). Tento prostor je naprosto dostačující pro vytvořené materiály, které zabírají přibližně 1/10 celkové kapacity.

9.1 Editor

Jedná se o jednoduchý Drag&Drop („táhni a pusť“) editor. Jednou z hlavních výhod je plně responzivní návrh. Stránky se tedy vždy přizpůsobí zařízení, na kterém je prohlížena, ať už se jedná o telefon, tablet nebo klasický monitor.

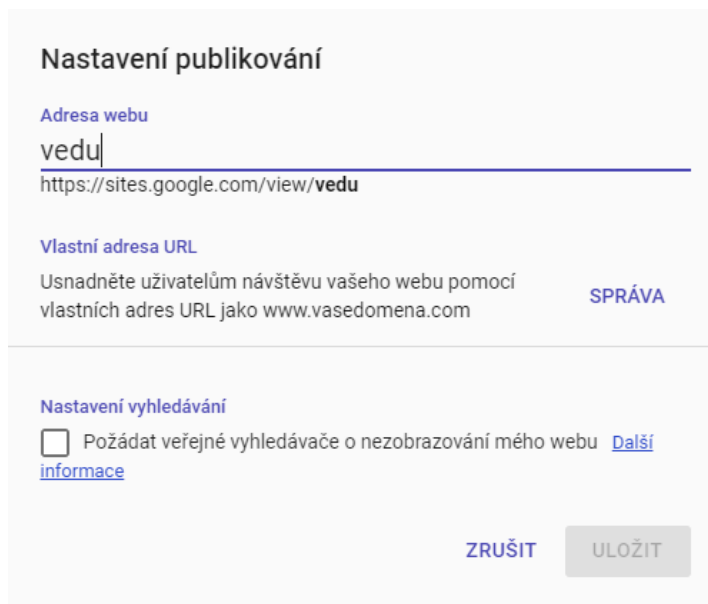
Uživateli umožňuje základní úpravy vzhledu a obsahu jednotlivých stránek i vytváření vlastní struktury odkazů. Složitější úpravy CSS Stylů ovšem v aplikaci umožněny nejsou.



Obrázek 22 Google Sites – hlavní okno editoru

9.2 Publikování

Dokončený web je možné publikovat přímo z aplikace Google Sites. V případě, že vlastníte doménu, je možné využít ji. Pokud doménu nevládníte, je nabízena pouze možnost upravit název v předem definované adrese <https://sites.google.com/view/nazev>.



Nastavení publikování

Adresa webu
vedu
<https://sites.google.com/view/vedu>

Vlastní adresa URL
Usnadněte uživatelům návštěvu vašeho webu pomocí vlastních adres URL jako www.vasedomena.com [SPRÁVA](#)

Nastavení vyhledávání
 Požádat veřejné vyhledávače o nezobrazování mého webu [Další informace](#)

[ZRUŠIT](#) [ULOŽIT](#)

Obrázek 23 Google Sites – Publikování

Pro lepší zapamatovatelnost a uživatelskou přívětivost je možné adresu modifikovat bezplatnými službami pro zkracování adres. Například pomocí webové stránky bit.ly, ow.ly nebo mnoha dalších.

Adresa <https://sites.google.com/view/vedu/domovsk%C3%A1-str%C3%A1nka> byla pomocí aplikace bitly zkrácena na: <http://bit.ly/veduc> ze které je návštěvník přesměrován na hlavní stránku.

ZÁVĚR

Hlavním cílem této práce bylo vytvořit materiály pro podporu výuky, včetně dalších podpůrných prostředků jako jsou prezentace a úkoly k vypracování. Jako platforma pro výuku byl zvolen projekt Arduino, který má pro daný účel mnoho předností jako je cena, možnosti a variabilita. I přes tyto výhody se ale Arduino ve výuce na SŠ používá poměrně málo.

Výběr stěžejních kapitol, které byly zpracovány, odpovídá společným tématům gymnaziální a obecné technické přípravy, konkrétně ŠVP Gymnázia Uherské Hradiště a ŠVP střední průmyslové školy Zlín. Oba dva jsou svou koncepcí zaměřeny na obecnou přípravu, jak humanitního, tak i technického směru. Pro každou z jednotlivých hodin byly vypracovány materiály pro učitele, materiály pro žáky, prezentace jako podpora výkladu a simulace většiny navrhovaných příkladů a úkolů. Velký důraz byl kladen na samotnou aktivitu žáků. Všechny tyto materiály byly umístěny na webovou stránku, tak aby byly k dispozici v ucelené formě na jednom místě.

Tato práce nepokrývá kompletní škálu připojitelných prvků ani všechny příkazy a možnosti, které je možné při práci využít. V tomto směru je určitě možné celou práci ještě rozšířit o složitější struktury a komponenty. Tato práce však poskytuje čtenáři odrazový můstek při návrhu vlastní výuky. Všechny navržené materiály by měly usnadnit implementaci do reálné praxe.

Díky široké podpoře komunity, různých způsobů zápisu kódu i zajímavému příslušenství pokrývá Arduino široké spektrum možností a využití, a to nejenom ve výuce. Věřím, že tato práce dokáže pomoci k přesvědčení učitelů k většímu rozšíření projektu Arduino do školních lavic.

SEZNAM POUŽITÉ LITERATURY

- [1] BARRAGÁN, Hernando. *Wiring: Prototyping Physical Interaction Design* [online]. Ivrea, 2004 [cit. 2019-04-07]. Dostupné z: http://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf. Master's thesis. Interaction Design Institute Ivrea. Vedoucí práce Massimo Banzi.
- [2] Introduction. *Arduino: Introduction* [online]. Turin: Arduino, 2018 [cit. 2019-01-16]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [3] BARRAGÁN, Hernando. The Untold History of Arduino. *The Untold History of Arduino* [online]. Bogota: Universidad de los Andes, 2016 [cit. 2019-04-07]. Dostupné z: <https://arduinhistory.github.io/>
- [4] About us: Credits. In: *Arduino* [online]. Turin: Arduino, 2019 [cit. 2019-02-27]. Dostupné z: <https://www.arduino.cc/en/main/credits>
- [5] Arduino Nano. In: *Arduino: Store* [online]. Somerville: Arduino, 2018 [cit. 2018-11-13]. Dostupné z: <https://store.arduino.cc/arduino-nano>
- [6] Arduino Micro. In: *Arduino: Store* [online]. Somerville: Arduino, 2018 [cit. 2018-11-13]. Dostupné z: <https://store.arduino.cc/arduino-micro>
- [7] Arduino Uno Rev3. In: *Arduino: Store* [online]. Somerville: Arduino, 2018 [cit. 2018-11-13]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [8] Arduino Mega 256 Rev 3. In: *Arduino: Store* [online]. Somerville: Arduino, 2018 [cit. 2018-11-13]. Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [9] Klon Arduino UNO R3 ATmega328P CH340 mini USB. In: *Arduino-shop: IOT Vývojové platformy* [online]. Havlíčkův Brod: Eclipsa, 2018 [cit. 2018-11-13]. Dostupné z: <https://arduino-shop.cz/arduino/1353-klon-arduino-uno-r3-atmega328p-ch340-mini-usb-1466635561.html>
- [10] Funduino výukový multi shield do škol pro Arduino. In: *Arduino-shop: Shield moduly* [online]. Havlíčkův Brod: Eclipsa, 2018 [cit. 2018-11-14]. Dostupné z: <https://arduino-shop.cz/arduino/1203-funduino-vyukovy-multi-shield-do-skol-1456079896.html>

- [11] Stavebnice LEGO Mindstorms 45544 EV3 Základní souprava. In: *Alza* [online]. Praha: Alza, 2019 [cit. 2019-04-07]. Dostupné z: <https://cdn.alza.cz/ImgW.ashx?fd=f4&cd=LO45544&i=1.jpg>
- [12] Aplikace: Ke stažení. In: *Lego: Mindstorms* [online]. Billund: Lego, 2019 [cit. 2019-04-07]. Dostupné z: <https://www.lego.com/cs-cz/mindstorms/downloads>
- [13] EV3 AND NXT. In: *Mindsensors* [online]. Baltimore: Mindsensors, c2005-2017 [cit. 2019-04-07]. Dostupné z: <http://www.mindsensors.com/37-ev3-and-nxt>
- [14] Inspire every child to create their best digital future: About the foundation. In: *Micro:bit* [online]. London: Micro:bit Educational Foundation, 2019 [cit. 2019-04-07]. Dostupné z: <https://microbit.org/about/>
- [15] Guide: Features. In: *Micro:bit* [online]. London: Micro:bit Educational Foundation, 2019 [cit. 2019-04-07]. Dostupné z: <https://microbit.org/guide/features/>
- [16] BBC micro:bit Starter kit. In: *Alza* [online]. Praha: Alza, 2019 [cit. 2019-04-07]. Dostupné z: https://www.alza.cz/bbc-microbit-starter-kit-d5479355.htm?kampan=adpla_vyrobci_Komponenty_programovatelne-stavebnice_c_1o1_1003736_MCB100a2_~32181666690~&gclid=EAIAIQobChMI3b2Qyqi-4QIVkuJ3Ch1zCgNWEAQYASABEGKPGfD_BwE
- [17] KONEČNÝ, Jan a Vilém VICHODIL. Paradigmata programování: 1A. In: *Phoenix* [online]. Olomouc: Univerzita Palackého v Olomouci Přírodovědecká Fakulta, b.r. [cit. 2019-04-14]. Dostupné z: https://phoenix.inf.upol.cz/esf/ucebni/uvod_para.pdf
- [18] SKOUPIL, David. Úvod do paradigmat programování. In: *Phoenix* [online]. Olomouc: Univerzita Palackého v Olomouci Přírodovědecká Fakulta, b.r. [cit. 2019-04-14]. Dostupné z: <https://phoenix.inf.upol.cz/esf/ucebni/pp1a.pdf>
- [19] SHEIKH, Ghazala a Noman ISLAM. A qualitative study of major programming languages: Teaching programming languages to computer science students. *Journal of Information & Communication Technology* [online]. Pakistan, 2016, **2016**(1), 24-34 [cit. 2019-04-14]. Dostupné z: https://www.researchgate.net/publication/306438454_A_qualitative_study_of_majo

r_programming_languages_teaching_programming_languages_to_computer_scienc
e_students

- [20] HORÁČKOVÁ, Hana. *Výuka programování v jazyce Python pro střední školy* [online]. Brno, 2014 [cit. 2019-04-14]. Dostupné z: <https://is.muni.cz/th/ljzfh/dp.pdf>. Diplomová práce. Masarykova univerzita v Brně Fakulta informatiky. Vedoucí práce Jaroslav Pelikán.
- [21] SAELI, Mara, Jacob PERRENET, Wim JOCHEMS a Bert ZWANEVELD. Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education* [online]. Vilnius, 2011, **2011**(1), 73–88 [cit. 2019-04-14]. Dostupné z: https://www.researchgate.net/publication/228951140_Teaching_Programming_in_Secondary_School_A_Pedagogical_Content_Knowledge_Perspective
- [22] FOJTÍK, Rostislav. Modern approaches to teaching programming. *Journal of Technology and Information Education: Časopis pro technickou a informační výchovu* [online]. 2013, **2013**(1), 58-62 [cit. 2019-04-12]. ISSN 1803-537X. Dostupné z: <https://jtie.upol.cz/pdfs/jti/2013/01/08.pdf>
- [23] About Swift: Platform Support. In: *Swift.org* [online]. Cupertino: Apple, 2019 [cit. 2019-04-13]. Dostupné z: <https://swift.org/about/#platform-support>
- [24] Rámcové vzdělávací programy. In: *Národní ústav pro vzdělávání* [online]. Praha: Národní ústav pro vzdělávání, 2019 [cit. 2019-04-15]. Dostupné z: <http://www.nuv.cz/t/rvp>
- [25] Rámcový vzdělávací program pro gymnázia. In: *Národní ústav pro vzdělávání* [online]. Praha: Národní ústav pro vzdělávání, 2019 [cit. 2019-04-15]. Dostupné z: <http://www.nuv.cz/file/159>
- [26] Školní vzdělávací program pro vyšší gymnázium: Labyrint světa II. In: *Gymnázium Uherské Hradiště: Dokumenty* [online]. Uherské Hradiště: Gymnázium Uherské Hradiště, 2017 [cit. 2019-04-18]. Dostupné z: https://www.gymnaziumuh.cz/dokumenty/svp_vg_GUH_18_19.pdf

- [27] Příloha školního vzdělávacího programu pro vyšší gymnázium: Labyrint světa II. In: *Gymnázium Uherské Hradiště: Dokumenty* [online]. Uherské Hradiště: Gymnázium Uherské Hradiště, 2017 [cit. 2019-04-18]. Dostupné z: https://www.gymnaziumuh.cz/dokumenty/svp_vg_prilohy_GUH_18_19.pdf
- [28] KALHOUS, Zdeněk a Otto OBST. *Školní didaktika*. Vyd. 2. Praha: Portál, 2009. ISBN 978-80-7367-571-4.
- [29] PASCH, Marvin. *Od vzdělávacího programu k vyučovací hodině*. Vyd. 2. Praha: Portál, 1998. ISBN 80-736-7054-2.
- [30] PRŮCHA, Jan. *Moderní pedagogika. 2.*, přeprac. a aktualiz. vyd. Praha: Portál, 2002. ISBN 80-717-8631-4.
- [31] MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012. ISBN 978-1449313876.
- [32] BANZI, Massimo a Michael SHILOH. *Getting started with Arduino*. Third edition. Sebastopol, CA: MakerMedia, 2015. ISBN 14-493-6333-4.
- [33] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.
- [34] Language Reference. In: *Arduino: Language Reference* [online]. Turin: Arduino, 2019 [cit. 2019-04-13]. Dostupné z: <https://www.arduino.cc/reference/en/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheets
GND	Ground
HID	Human interface device
IDE	Integrated Development Environment
LCD	Liquid crystal display
LED	Light – Emitting Diode
PIR	Passive infrared sensor
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer
RVP	Rámcový vzdělávací program
ŠVP	Školní vzdělávací program
USB	Universal Serial Bus

SEZNAM OBRÁZKŮ

Obrázek 1 Schéma umístění základních prvků na desce Arduino UNO.....	12
Obrázek 2 Arduino Nano [5].....	13
Obrázek 3 Arduino Micro [6].....	14
Obrázek 4 Arduino UNO [7].....	14
Obrázek 5 Arduino Mega [8]	15
Obrázek 6 Arduino UNO klon [9].....	16
Obrázek 7 shield školní rozšíření pro arduino [10].....	17
Obrázek 8 Arduino IDE.....	19
Obrázek 9 Ardublockly	20
Obrázek 10 Autodesk Tinkercad.....	21
Obrázek 11 LEGO Mindstorms – education [11]	22
Obrázek 12 zápis algoritmu v programu Lego Mindstorms.....	23
Obrázek 13 Micro:bit přehled komponent [15].....	24
Obrázek 14 zápis algoritmu v aplikaci MakeCode Editor.....	24
Obrázek 15 dotazník – zastoupení typů středních škol	27
Obrázek 16 dotazník – zaměření středních škol.....	28
Obrázek 17 dotazník – vyučované programovací jazyky.....	29
Obrázek 18 dotazník – využití platformy Arduino ve výuce	29
Obrázek 19 dotazník – důvody nevyužití platformy Arduino ve výuce	30
Obrázek 20 dotazník – konkrétní příklady využití ve výuce.....	30
Obrázek 21 ukázka prostředí Prezentace Google.....	56
Obrázek 22 Google Sites – hlavní okno editoru.....	57
Obrázek 23 Google Sites – Publikování.....	58

SEZNAM TABULEK

Tabulka 1 Srovnání jednotlivých Arduino desek	15
Tabulka 2 ŠVP GUH – Informační technologie [26]	39
Tabulka 3 ŠVP GUH – Programování pro 3. ročník	39
Tabulka 4 ŠVP SPSZL – Technické lyceum – programování pro 2. ročník	41
Tabulka 5 seznam zpracovaných témat a jejich předpokládané časové rozvržení	49

SEZNAM PŘÍLOH

- P I Dotazník
- P II Ukázkový materiál pro učitele

PŘÍLOHA P I: DOTAZNÍK.

14. 4. 2019

Programování na platformě Arduino na středních školách v ČR

Programování na platformě Arduino na středních školách v ČR

Dobrý den,

jmenuji se Viktor Vráblík, jsem studentem Fakulty Aplikované Informatiky Univerzity Tomáše Bati ve Zlíně. Obracím se na Vás s prosbou vyplnění krátkého dotazníku.

Diplomová práce se zabývá vytvořením přípravy hodin a výukových materiálů pro programování na platformě Arduino.

Jako poděkování za Váš čas, Vám tyto materiály po odevzdání diplomové práce zašlu na email, který je možné uvést na konci dotazníku. Tyto materiály Vám mohou posloužit jako komplexní základ případné výuky.

Ukázky materiálů: <https://drive.google.com/drive/folders/1Tzo-VjvOV1kUAKEvUXTYzwwFXmSDnQZG?usp=sharing>
Simulace úkolu 1: <https://www.tinkercad.com/things/9N2zXsQiyvS>

Velmi Vám děkuji za čas věnovaný dotazníku.
S pozdravem a přáním hezkého dne Bc. Viktor Vráblík.

***Povinné pole**

1. Samosprávný kraj školy *

Označte jen jednu elipsu.

- Hlavní město Praha
- Jihočeský kraj
- Jihomoravský kraj
- Karlovarský kraj
- Kraj Vysočina
- Královéhradecký kraj
- Liberecký kraj
- Moravskoslezský kraj
- Olomoucký kraj
- Pardubický kraj
- Plzeňský kraj
- Středočeský kraj
- Ústecký kraj
- Zlínský kraj

2. Zřizovatel školy *

Označte jen jednu elipsu.

- Veřejná škola (stát, kraj, obec)
- Soukromá škola
- Církevní škola

3. Typ školy *

Uveďte typ školy. Možno vybrat více odpovědí.
Zaškrtněte všechny platné možnosti.

- Gymnázium
 Střední odborná škola
 Střední odborné učiliště

4. Zaměření

V případě, že jste odpověděli na předchozí otázku jinak než Gymnázium, uveďte prosím v další odpovědi zaměření vaší školy. Možno vybrat více odpovědí
Zaškrtněte všechny platné možnosti.

- Obchod a Služby
 Strojírenství
 Elektrotechnika
 Informatika
 Stavebnictví
 Jiné

5. Výuka programování *

Zde prosím vyplňte zda na Vaší škole probíhá výuka programování (v jakékoli podobě).
Označte jen jednu elipsu.

- Ano - jako povinný předmět (hlavní předmět oboru)
 Ano - jako nepovinný předmět (v rámci seminářů, nebo kroužků)
 Ne *Přeskočte na otázku 18.*

Výuka Programování**6. Programovací jazyky**

V prvním sloupci vyberte programovací jazyk, který slouží jako úvodní jazyk do problematiky programování. V druhém sloupci vyberte všechny jazyky, které se na Vaší škole vyučují (nejen jako hlavní vyučovací předmět, ale i jako volitelný seminář, nebo kroužek).
Zaškrtněte všechny platné možnosti.

	První programovací jazyk	Na naší škole se vyučuje
Pascal	<input type="checkbox"/>	<input type="checkbox"/>
Delphi	<input type="checkbox"/>	<input type="checkbox"/>
C	<input type="checkbox"/>	<input type="checkbox"/>
C++	<input type="checkbox"/>	<input type="checkbox"/>
C#	<input type="checkbox"/>	<input type="checkbox"/>
Java	<input type="checkbox"/>	<input type="checkbox"/>
Python	<input type="checkbox"/>	<input type="checkbox"/>
Arduino (Wire)	<input type="checkbox"/>	<input type="checkbox"/>
Scratch nebo jiný grafický programovací jazyk	<input type="checkbox"/>	<input type="checkbox"/>
Assembler	<input type="checkbox"/>	<input type="checkbox"/>
Jiné	<input type="checkbox"/>	<input type="checkbox"/>

7. Arduino - povědomí *

O platformě Arduino (nebo o kompatibilním klonu) a jejím využití v praxi jsem již slyšel.
Označte jen jednu elipsu.

- Ano o platformě Arduino jsem slyšel.
- Ne o platformě Arduino jsem neslyšel.

8. Arduino - zkušenost *

S platformou Arduino (nebo s kompatibilním klonem) mám osobní zkušenost mimo výuku.
Označte jen jednu elipsu.

- Ano s platformou Arduino mám osobní zkušenost.
- Ne s platformou Arduino nemám osobní zkušenost.

9. Arduino - ve výuce *

Platformu Arduino (nebo kompatibilní klon) využívám v praktické výuce.
Označte jen jednu elipsu.

- Ano využívám platformu Arduino ve výuce. *Přeskočte na otázku 11.*
- Ne nevyžívám platformu Arduino ve výuce. *Přeskočte na otázku 10.*

Arduino ve výuce

10. Arduino - ve výuce *

Hlavní důvod nezačlenění platformy Arduino (nebo kompatibilního klonu) do výuky.
Označte jen jednu elipsu.

- Příliš finančně nákladné
- Využívám jinou alternativu (např. LEGO Mindstorm, RaspberryPi, starší mikroprocesory...)
- Přílišná složitost
- Nevyhovuje potřebám výuky
- Málo dostupných materiálů

Přeskočte na otázku 18.

Arduino ve výuce

11. Arduino - Konkrétní předměty *

Na kterých předmětech pracujete s platformou Arduino (nebo s kompatibilním klonem) v rámci výuky.

Zaškrtněte všechny platné možnosti.

- Programování - algoritmizace
- Programování mikropočítačů
- Elektrotechnika
- Robotika
- Jiné: _____

12. Arduino - Přednosti *

Jaké jsou hlavní přednosti platformy Arduino (nebo kompatibilního klonu).
Označte jen jednu elipsu.

- Cenová dostupnost
- Uživatelská komunita
- Jednoduchost

13. Arduino - Zápís kódu *

Který způsob zápisu programu ve výuce převažuje
Označte jen jednu elipsu.

- Zápis kódu - Arduino IDE
- Zápis kódu - VisualCode, Atmel Studio ...
- Grafické programování - Ardublock, Visiuno, ...

14. Arduino - Simulátory *

Víte o tom, že existují online simulátory, které je možné využít jako komplexní alternativu?
Označte jen jednu elipsu.

- Ano vím o existenci simulátorů, ale ve výuce je nevyžívám.
- Ano vím o existenci simulátorů, a ve výuce je využívám.
- Ne nevím o existenci simulátorů.

O vás

15. Věk

Zde prosím uveďte Váš věk. Odpověď uvádějte jako číselný údaj.

16. Pohlaví

Označte jen jednu elipsu.

- Muž
- Žena

17. Délka praxe

Zde prosím uveďte délku vaší praxe. Odpověď uvádějte jako číselný údaj jako celé číslo.

Přeskočte na otázku 18.

Poděkování

Děkuji za čas věnovaný vyplnění dotazníku.

Pokud máte zájem o zaslání vypracovaných materiálů, uveďte prosím v další otázce Vaši emailovou adresu.

s pozdravem Bc. Viktor Vráblík

18. Kontaktní email

19. Výsledky dotazníku

Označte jen jednu elipsu.

Ano - chci zaslat i výsledky statistického šetření

20. Doplnění

Chybělo Vám něco v dotazníku, co by jste chtěl/a doplnit?

PŘÍLOHA P II: UKÁZKOVÝ MATERIÁL PRO UČITELE

Vráblík Viktor, 2019: bit.ly/veduc – CC BY-NC-SA

Arduino – Cykly

Téma vyučovací hodiny

Vymezení základního pojmu cyklu. Otočka cyklu, řídicí podmínka a řídicí proměnná. Vysvětlení základních typů cyklů: cyklus s podmínkou na začátku, cyklus s podmínkou na konci. Cyklus for. Praktické využití na konkrétních příkladech.

Cíl hodiny

- Žák umí vlastními slovy vysvětlit pojem cyklus.
- Žák umí definovat základní pojmy: řídicí podmínka, řídicí proměnná a otočka cyklu
- Žák dokáže popsat princip jednotlivých druhů cyklů.
- Žák porovná výhody a nevýhody jednotlivých druhů cyklů.
- Žák je schopen vysvětlit rozdíl mezi příkazem *break* a *continue*, na konkrétních příkladech.
- Žák je schopen analyzovat ukázkové příklady, je schopen popsat vlastními slovy jednotlivé kroky
- Žák dokáže použít získané znalosti na jednoduchých algoritmických úlohách
- Žák dokáže řešení úloh zapsat je v jazyku Arduino.

Klíčové kompetence

- Kompetence k řešení problémů
- Kompetence k učení
- Kompetence pracovní

Organizační formy výuky

- Hromadná a frontální výuka
- Skupinová výuka – samostatná práce na úkolech. je možné organizovat v rámci dvojic s odlišně obtížným zadáním.

Metody výuky

- Metody informačně receptivní – prezentace informace učitelem
- Metody reproduktivní – diskuse nad řešenými příklady a úkoly
- Metody problémového výkladu – samostatná práce na úkolech

Použité didaktické prostředky

- Výukové prostory a jejich vybavení: Počítačová učebna
- Technické výukové prostředky: Audiovizuální technika – dataprojektor tabule
- Učební pomůcky: Textové pomůcky – pracovní materiály
- Speciální pomůcky: Žákovské experimentální pomůcky – deska Arduino a základní komponenty (nebo simulátor stejné funkcionality). Seznam externích komponent:
 - o V tomto materiál je využita pouze deska Arduino

Časové rozvržení

Čas [min]	Obsah
0-5	Začátek hodiny, zápis do třídní knihy.
5-10	Cykly obecně
10-15	While cyklus
15-20	
20-25	Do – while cyklus
25-30	
30-35	For cyklus
35-40	
40-45	
0-5	Příklad 1 – komentovaný příklad
5-10	Diskuse nad programem
10-15	
15-20	Úkol 1 (+ případně úkol 2 podle volného času)
20-25	
25-30	
30-35	Samostatná práce
30-35	Diskuse nad řešením
35-40	
40-45	Závěr hodiny, rekapitulace získaných vědomostí

Výuka

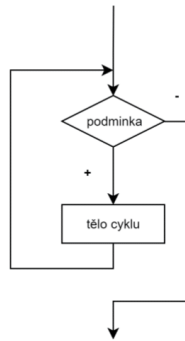
Cykly slouží ke kontrolovanému opakování určité části programu. V této lekci si vysvětlíme tři základní struktury. While, Do-While a For cyklus. Vysvětlíme rozdíly a příklady použití.

Cykly

Pokud v rámci programu chceme vykonávat nějakou část opakovaně, nebylo by příliš efektivní psát přímo do kódu, 100x tu samou část programu. Nejenom že by nám narůstala velikost samotného programu, ale celý zápis by se stal velmi nepřehledný. Cykly byly implementovány právě z tohoto důvodu. Umožňují nám opakovat část kódu, v závislosti na splnění podmínce nebo počtu opakování.

While cyklus

While cyklus neboli cyklus s podmínkou na začátku, vykonává tělo cyklu pouze pokud je splněna podmínka. Tato podmínka se kontroluje vždy před každou otočkou cyklu (Otočka cyklu je jedno vykonání těla cyklu, tedy příkazů, které jsou uvedeny uvnitř). Graficky by se celý tento děj dal v jazyku vývojových diagramů zakreslit takto:



V případě, že je splněná podmínka, přechází se k vykonávání těla cyklu. Na konci se opět přejde k vyhodnocení podmínky, pokud je výraz vyhodnocen jako pravda (podmínka je splněna), přejde se k další iteraci cyklu. Pokud ne pokračuje se v kódu, který je uveden za koncem celého cyklu. [1]

V programovacím jazyku by zápis tohoto cyklu vypadal následovně:

```
while(podmínka)
{
    //tělo cyklu
}
```

Pokud bychom chtěli například k číslu n , které zadá uživatel přičítat 5, do doby, dokud jeho hodnota nepřesáhne 50, mohl by zápis kódu vypadat následovně.

```
while(n<50)           //dokud platí že hodnota proměnné n je menší než 50
{
    Serial.println(n); //vypiš hodnotu proměnné n na sériový monitor
    n=n+5;             //zvyš hodnotu proměnné n o 5
}
```

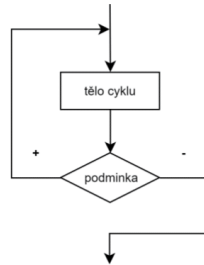

Nejprve je nutné vymyslet podmínku, podle které se cyklus bude řídit. Jelikož chceme postupně přičítat číslo 5 k zadanému číslu n budeme v řídicí podmínce kontrolovat, zda je hodnota proměnné n menší než 50.

V těle cyklu nejprve vypíšeme hodnotu proměnné n a poté ji zvětšíme o 5. Následně se vracíme na začátek cyklu a ověřujeme platnost podmínky. Toto opakujeme do té doby, dokud v těle cyklu nezvětšíme proměnnou na hodnotu větší než 50, v tento okamžik už není splněná podmínka a tělo cyklu se již nevykoná. Poté se začne vykonávat část kódu, která následuje za cyklem.

Pokud by uživatel zadal počáteční hodnotu větší než 50 do těla cyklu by se nepřešlo a cyklus by rovnou skončil.

Do-While

Další varianta cyklu je cyklus Do-While neboli cyklus s podmínkou na konci. Oproti While cyklu, u kterého se tělo cyklu nemuselo vykonat ani jednou, je tady tomu jinak. V případě Do-While cyklu je tělo cyklu vykonáno vždy alespoň jednou. Podmínka je kontrolována až po vykonání těla cyklu. [2]



V programovacím jazyku zapsáno jako:

```
do
{
    //tělo cyklu
}
while (podmínka)
```

Podobně, pokud bychom chtěli k číslu n přičítat hodnotu 5 dokud je hodnota proměnné n menší než 50 (jinak řečeno, dokud číslo n není větší než 50), mohl by zápis v kódu vypadat takto.

```
do
{
    Serial.println(n);    //vypiš hodnotu proměnné n na sériový monitor
    n=n+5;                //zvyš hodnotu proměnné n o 5
}
while (i<5)
```

Při konstrukci cyklu je opět nejprve nutné promyslet řídicí podmínku. V tomto konkrétním případě je možné zapsat podmínku stejně jako u while cyklu. Tedy dělej tělo cyklu, dokud je hodnota v proměnné n menší než 50.

Nejde však tvrdit, že budou podmínky v případě záměny obou cyklů totožné. Z větší části případu se řídicí podmínka bude u obou cyklů lišit i tehdy pokud bychom je využili na stejný problém.

V těle cyklu poté vždy jen vypíšeme hodnotu proměnné `n` na seriový monitor a zvýšíme hodnotu proměnné `n` o 5.

P Poznámka

Obecně platí, že vše, co se dá zapsat pomocí cyklu *while*, lze zapsat i cyklem *do-while*. Záleží pak na konkrétním problému a uvážení programátora.

For cyklus

Předchozí dva cykly se obecně používají v případech, kdy nevíme kolikrát se tělo cyklu bude opakovat. Pokud už dopředu víme, kolikrát chceme opakovat tělo cyklu, je nejjednodušším řešením, využít cyklu `for`. Přesný počet opakování, řídí tzv. řídicí proměnná – celočíselná hodnota. Hodnota této proměnné se v každé otočce cyklu mění podle příkazu, který je uveden při inicializaci. [3]

Rovnou si vysvětleme obecný zápis v programovacím jazyku:

```
for (řídicí proměnná; podmínka; příkaz)
{
    //tělo cyklu
}
```

Řídicí proměnné je na začátku potřeba nastavit počáteční hodnotu (nejčastěji 0, ale není to pravidlem, záleží na konkrétním příkladu, můžeme klidně počáteční hodnotu změnit). Podmínka, která se kontroluje vždy na začátku každé otočky cyklu. A příkaz, který se zavolá nad řídicí proměnnou vždy na konci těla cyklu.

Pokud bychom chtěli vypsat čísla od 0 do 4, tedy vykonat tělo cyklu pětkrát, zápis pomocí `for` cyklu by vypadal následovně:

```
for (int i = 0; i < 5; i++)
{
    Serial.println(i); // výpis hodnoty proměnné i
}
```

Řídicí proměnné je nastavena počáteční hodnota 0 (všimněte si, že řídicí proměnná je přímo v zápisu cyklu deklarována). Podmínka je kontrolována na začátku otočky cyklu. Příkaz se provede na konci otočky.

Jelikož začínáme od nuly, je podmínka `i < 5` splněna, přechází se tedy k vykonávání těla cyklu. Poté, co se vykoná poslední příkaz těla cyklu je vykonán příkaz uvedený v inicializaci cyklu. V našem případě tedy `i++`.

Jakmile je podmínka cyklu nesplněna, řídicí proměnná `i` je větší nebo rovna 5, je provádění `for` cyklu ukončeno a přejde se k přecházení další části kódu po cyklu.

Řídicí příkazy

Chod cyklu jde mimo kontrolní podmínky na začátku, případně na konci cyklu, řídit také pomocí dalších příkazů přímo uvnitř cyklu. Konkrétně se jedná o příkazy *break* a *continue*.

Příkaz break

Příkaz *break*, kdykoli je vykonán, ukončí cyklus, uvnitř kterého je příkaz *break* volán. [4]

Pokud bychom chtěli například načíst od uživatele 5 čísel, které bychom chtěli sečíst. V případě, že by uživatel zadal 0, okamžitě bychom celý cyklus načítání ukončili.

```
for (int i = 0; i<4; i++)
{
    while (Serial.available() == 0) {}
    promenna = Serial.parseInt();

    if(promenna == 0) {break;}
    else {suma = suma + promenna;}
}
```

Příkaz continue

Příkaz *continue*, kdykoliv je vykonán, přeskočí jednu iteraci cyklu. Jinak řečeno ukončí vykonávání těla cyklu, a vrátí se opět na začátek a začne tělo cyklu vykonávat znovu. [4]

Pokud bychom například chtěli vykonat část cyklu pouze tehdy kdy by řídicí proměnná byla sudá, mohlo by řešení za pomoci příkazu *continue* vypadat takto.

```
for (int i = 0; i<4; i++)
{
    if (i %2!= 0)           //pokud je zbytek po celočíselném dělení různý od nuly – číslo je liché
    {continue;}           //přeskoč jednu iteraci cyklu

    //zbytek těla cyklu;
}
```

Příkaz modulo vrací hodnotu zbytku po celočíselném dělení. V kódu se zapisuje symbolem %. (5%4 == 1; 10%2 == 0;)

P Poznámka

Pokud voláme řídicí příkaz *break* nebo *continue* v rámci vnořeného cyklu. Ukončí se (nebo přeskočí) pouze úroveň, ve které je příkaz volán. [5]

Příklady

P Příklad 1

Načtěte od uživatele číselnou hodnotu. Tu přiřadte do proměnné. Pomocí opakovaného přičítání spočítejte druhou mocninu tohoto čísla. Tuto hodnotu vypište na výstup.

V tomto příkladu není nutné připojit žádné externí komponenty, proto zde není uvedeno schéma zapojení

Násobení se obecně dá napsat jako opakované přičítání. Například výraz $2 \cdot 3$ se dá napsat jako $2+2+2$ nebo jako $3+3$. U mocniny tomu není jinak. Například na 3^2 se můžeme dívat jako na $3 \cdot 3$ nebo jako $3+3+3$. Tedy třikrát přičtená hodnota tři. Obecně pokud bychom chtěli vypočítat druhou mocninu z čísla n , můžeme se na celou operaci dívat jako na n -krát přičtená hodnota n .

```
int cislo;           //deklarace celočíselné proměnné číslo
int mocnina;        //deklarace celočíselné proměnné výsledek

void setup()
```

```

{
    Serial.begin(9600); // nastavení rychlosti sériové komunikace
}

void loop()
{
    Serial.println("Zadejte cislo na umocneni");
    while (Serial.available() == 0) {} //čekání na vstup na seriové lince
    cislo = Serial.parseInt(); //načti hodnotu do proměnné prom

    mocnina = 0; //vynulování proměnné mocnina
    for(int i = 0; i<cislo; i++) //dokud je řídicí proměnná menší než hodnota prom. číslo
    {
        mocnina = mocnina + cislo; //přičti k výsledku hodnotu čísla
    }

    Serial.println(mocnina); //vypiš hodnotu výsledku
}

```

Úkoly

U Úkol 1

Načtete od uživatele dvě číselné hodnoty. Dělenec a dělitel. Pomocí opakovaného odečítání spočítejte zbytek po celočíselném dělení. Hodnotu vypíšete na sériovou linku.

V tomto příkladu není nutné připojit žádné externí komponenty, proto zde není uvedeno schéma zapojení.

```

int delenec; //deklarace celočíselné proměnné delenec
int delitel; //deklarace celočíselné proměnné delitel

void setup()
{
    Serial.begin(9600); // nastavení rychlosti sériové komunikace
}

void loop()
{
    Serial.println("Zadejte hodnotu delence");
    while (Serial.available() == 0) {} //čekání na vstup na seriové lince
    delenec = Serial.parseInt(); //přiřaď hodnotu do proměnné delenec

    Serial.println("Zadejte hodnotu delitele");
    while (Serial.available() == 0) {} //čekání na vstup na seriové lince
    delitel = Serial.parseInt(); //přiřaď hodnotu do proměnné delitel

    while(delenec>=delitel)
    {
        delenec = delenec - delitel;
    }
    Serial.println(delenec); //vypiš hodnotu výsledku
}

```

U Úkol 2

Načtete od uživatele celočíselnou hodnotu. Ověřte, že je kladná a větší než 0. Pokud ano, zablikejte LED na Arduino desce tolikrát, kolik je hodnota čísla zadaného uživatelem. Využijte vhodný cyklus.

V tomto příkladu není nutné připojit žádné externí komponenty, proto zde není uvedeno schéma zapojení.

```
int cislo; //deklarace celočíselné proměnné číslo

void setup()
{
  Serial.begin(9600); // nastavení rychlosti sériové komunikace
  pinMode(13, OUTPUT); //nastavíme pin 13, jako výstupní
}

void loop()
{
  Serial.println("Zadejte číslo");
  while (Serial.available() == 0) {} //čekání na vstup na seriové lince
  cislo = Serial.parseInt(); //načti hodnotu do proměnné prom

  for(int i = 0; i<cislo; i++) //dokud je řídicí proměnná menší než hodnota prom. číslo
  {
    digitalWrite(13,HIGH); //Rozsviť LED na pinu 13
    delay(1000); //počkej 1s
    digitalWrite(13,LOW); //Zhasni LED na pinu 13
    delay(1000); //počkej 1s
  }
}
```

Literatura

- [1] While: Control Structure. In: *Arduino: Structure* [online]. Turin: Arduino, 2019 [cit. 2019-03-25]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/while/>
- [2] Do...while: Control Structure. In: *Arduino: Structure* [online]. Turin: Arduino, 2019 [cit. 2019-03-25]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/dowhile/>
- [3] For: Control Structure. In: *Arduino: Structure* [online]. Turin: Arduino, 2019 [cit. 2019-03-25]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/for/>
- [4] MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012. ISBN 978-1449313876.

[5] BANZI, Massimo a Michael SHILOH. *Getting started with Arduino*. Third edition. Sebastopol, CA: MakerMedia, 2015. ISBN 14-493-6333-4.

[6] DigitalWrite(). In: *Arduino: Language Reference* [online]. Turin: Arduino, 2018 [cit. 2019-01-18]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>