



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v ČB

Pedagogická fakulta

Katedra informatiky

Programování Micro:bitu v blokově orientovaném prostředí Programming of Micro:bit in block-oriented environment

Diplomová práce

Vypracoval: Jan Pršala

Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice

2021

Prohlášení

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 16. dubna 2021.

.....

Jan Pršala

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan PRŠALA**
Osobní číslo: **P19287**
Studijní program: **N7503 Učitelství pro základní školy**
Studijní obor: **Učitelství matematiky pro 2. stupeň základních škol**
Učitelství informatiky pro 2. stupeň základních škol
Téma práce: **Programování Micro:bitu v blokově orientovaném prostředí**
Zadávací katedra: **Katedra informatiky**

Zásady pro vypracování

Kvalifikační práce je věnována výuce algoritmizace na druhém stupni ZŠ pomocí programovatelné desky Micro:bit. Pro ni je v projektu PRIM vytvořena učebnice v jazyce Python. Učebnice je vytvořena v licenci CC-BY-SA.

V teoretické části práce student porovná alespoň dvě bloková prostředí, vhodná k programování Micro:bitu, z pohledu jejich možností tuto desku programovat. Stručně popíše způsob programování Micro:bitu ve vybraném blokovém prostředí a také rozdíly mezi programováním této desky v tomto prostředí a v Pythonu.

V praktické části student vytvoří verzi existující učebnice pro blokově orientované prostředí včetně obrázků. Prověří, které z úloh ve zdrojové učebnici nelze převést do blokového prostředí, a pro takové úlohy navrhne náhradu. Odstraní nebo nahradí nerelevantní texty vztahující se k původnímu jazyku Python. Takto upravenou učebnici publikuje pod stejnou licenci.

Rozsah pracovní zprávy: **60**
Rozsah grafických prací: **CD ROM**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

1. PECH, J., & NOVÁK, M. Robotika: učebnice pro střední školy – Micro:Bit. Dostupné z: <https://imysleni.cz/ucebnice/robotika-ucebnice-pro-stredni-skoly-micro-bit>
2. Micro:bit ve výuce (2019). Dostupné z: <https://www.microbiti.cz/>
3. HAVÍŘOVÁ, B. (2019). BBC Micro:bit ve škole. e-Mole: časopis o výuce nejen s digitálními technologiemi. Tomáš Feltl – TFSOFT. Dostupné z: <https://www.e-mole.cz/clanek/bbc-microbit-ve-skole>
4. SENTANCE, S., WAITE, J., HODGES, S., MACLEOD, E., & YEOMANS, L. E. (2017). „Creating Cool Stuff“ – Pupils’ experience of the BBC micro:bit. In Proceedings of the 48th ACM Technical Symposium on Computer Science Education: SIGCSE 2017. Dostupné z: <https://doi.org/10.1145/3017680.3017749>
5. SENEVIRATNE, P. (2018). Beginning BBC micro:bit: a practical introduction to Micro:bit development. New York, NY: Apress.
6. GATENBY, J., & GATENBY J. (2017). Python Coding on the BBC Micro:Bit. Londýn: Bernard Babani Publishing.
7. VANÍČEK J., NAGYOVÁ I., & TOMCSÁNYIOVÁ M. Programování ve Scratchi pro 2. stupeň základní školy. Dostupné z: <https://imysleni.cz/ucebnice/programovani-ve-scratchi-pro-2-stupen-zakladni-skoly>
8. DENNING P., & TEDRE M. (2019). Computational Thinking. Cambridge, Massachusetts: The MIT Press.

9. KRAUSS, J., & PROTTSMAN, K. (2017). Computational thinking and coding for every student: the teacher's getting-started guide. Thousand Oaks, California: SAGE Publishing.
10. MANNAPPERUMA P. (2017). Start your micro:bit journey: with MakeCode & MU Editor.
11. KHINE, M., (2018). Computational thinking in the STEM disciplines: foundations and research highlights. New York, NY: Springer Berlin Heidelberg.
12. MORRIS D., (2017). Teaching Computational Thinking and Coding in Primary Schools. Thousand Oaks, California: SAGE Publishing

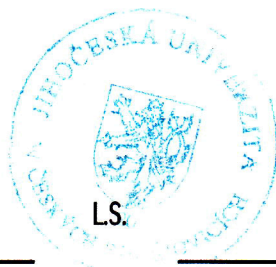
Vedoucí diplomové práce: **doc. PaedDr. Jiří Vaníček, Ph.D.**
Katedra informatiky

Datum zadání diplomové práce: **6. listopadu 2019**

Termín odevzdání diplomové práce: **30. dubna 2021**



doc. RNDr. Helena Koldová, Ph.D.
děkanka



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

Anotace

Tato diplomové práce se zabývá vytvořením učebnice programování pro programovatelnou desku micro:bit v blokovém prostředí pro druhý stupeň ZŠ. Práce vychází z učebnice robotiky pro střední školy z projektu PRIM. Originální učebnice je určena pro žáky středních technických škol a zabývá se programováním micro:bitu v Pythonu.

V teoretické části práce je popsána programovatelná deska micro:bit, její funkce, historie a využití ve výuce. Kromě toho jsou zde porovnány různá bloková prostředí a jejich vhodnost při programování micro:bitu. Také je popsán rozdíl mezi programováním micro:bitu v Pythonu a vybraným blokovým prostředím.

V praktické části práce jsou analyzovány všechny úlohy v původní učebnici. Dochází k prověření, které zdrojové úlohy nelze převést do blokového prostředí. Pro tyto úlohy jsou navrženy náhradní úlohy.

Samotná učebnice je rozdělena do několik kapitol. Každá kapitola popisuje jednu funkci micro:bitu, například práce s tlačítky, zvukem nebo vzájemnou komunikaci dvou a více micro:bitů. Žáci si pomocí programování jednoduchých programů tyto funkce osvojí. Učebnice je volně dostupná na internetu pod stejnou licencí jako originální učebnice.

Klíčová slova

micro:bit, robotika, infromatické myšlení, programování, algoritmizace

Abstract

This thesis is focusing on creating a textbook designed for secondary schools about programming a programmable board micro:bit in a block environment. The thesis is based on a similar textbook designed for high schools from a project PRIM. This original textbook is designed for technical high schools and deals with programming micro:bit in Python.

The programmable board micro:bit, her functions, history and usability in schools are described in the theoretical part. Aside from that, various block environments and their usability in programming micro:bit are also described there. Programming micro:bit in a selected environment is then compared to programming it in Python.

All exercises from the original textbook are analysed in the practical part of the thesis. Exercises are tested and it's decided if they can be recreated in the selected environment. For those exercises that can't be recreated, new exercises are created instead of them.

The textbook itself is divided into several chapters. Each chapter describes one function of micro:bit, for example how buttons, sound or communication between two or more micro:bits work. Pupils will learn about those functions via elementary exercises. The textbook is free for download on the internet under the same license as the original textbook.

Keywords

micro:bit, robotics, computational thinking, programming, algorithmization

Poděkování

Rád bych poděkoval panu docentovi Jiřímu Vaničkovi za ochotu při vedení mé diplomové práce, za cenné rady, nápady a připomínky. Také děkuji panu doktorovi Jiřímu Pechovi za povolení využít jeho učebnici jako základ pro tuto práci a za jeho rady, nápady a připomínky, které mi k učebnici poskytl. Zároveň bych chtěl poděkovat své přítelkyni Denise Janů za kritické připomínky k vytvořeným úlohám.

Obsah

1	Úvod	11
1.1	Cíl práce	11
1.2	Metoda práce	12
2	Informatické myšlení	14
3	Programovací deska BBC micro:bit	17
3.1	Obecné informace	17
3.2	Historie	17
3.3	Bloková prostředí pro programování Micro:bitu	18
3.3.1	Scratch	19
3.3.2	Microsoft MakeCode	19
3.3.3	MicroBlocks	20
3.3.4	Vittascience	21
3.3.5	Shrnutí	22
3.4	Porovnání Microsoft MakeCode a Mu Python Editoru	23
4	Překlad prostředí Microsoft MakeCode	25
5	Přehled úloh v původní učebnici	27
5.1	Kapitola 1 - LED displej	27
5.1.1	Hodina 1	27
5.1.2	Hodina 2	27
5.1.3	Hodina 3	28
5.1.4	Hodina 4-5	29
5.2	Kapitola 2 - Tlačítka	29
5.2.1	Hodina 1	30
5.3	Kapitola 3 - Zvuk	30
5.3.1	Hodina 1	30
5.3.2	Hodina 2	31

5.4	Kapitola 4 - Poloha	31
5.4.1	Hodina 1	31
5.4.2	Hodina 2	31
5.4.3	Hodina 3	32
5.4.4	Hodina 4	32
5.5	Kapitola 5 - Síť	32
5.5.1	Hodina 1	32
5.5.2	Hodina 2	33
5.5.3	Hodina 3	33
5.5.4	Hodina 4	33
5.6	Kapitola 6 - Periférie	33
5.6.1	Hodina 1	33
5.6.2	Hodina 2	33
5.6.3	Hodina 3	34
5.7	Závěrečný projekt	34
5.8	Žákovské listy	34
5.9	Samostudium	34
6	První verze nové učebnice	35
6.1	Obecné změny v učebnici	35
6.2	Nové úlohy	35
6.3	Popis změn po kapitolách	36
6.3.1	Kapitola 1	37
6.3.2	Kapitola 2	37
6.3.3	Kapitola 3	38
6.3.4	Kapitola 4	38
6.3.5	Kapitola 5	39
6.3.6	Kapitola 6	40
6.3.7	Závěrečný projekt	40
6.3.8	Žákovské listy	41

6.3.9	Samostudium	41
7	Finální verze nové učebnice	42
7.1	Důvody pro další změnu učebnice	42
7.2	Obecné změny v učebnici	42
7.3	Popis změn po kapitolách	43
7.3.1	Kapitola 1	43
7.3.2	Kapitola 2	44
7.3.3	Kapitola 3	44
7.3.4	Kapitola 4	44
7.3.5	Kapitola 5	45
7.3.6	Kapitola 6	46
7.3.7	Závěrečný projekt	46
7.3.8	Žákovské listy	47
7.4	Vydání učebnice	47
8	Závěr	49
	Seznam použité literatury a zdrojů	51
	Seznam obrázků	53
A	Příloha 1: Metodické listy učebnice pro učitele	54
B	Příloha 2: Žákovské listy učebnice	132

1 Úvod

V současné době dochází ve vzdělávací oblasti Informatika a informační a komunikační technologie k velkým změnám. Klade se větší důraz zejména na samostatnou práci žáků a rozvoj infortatického myšlení. Jedna z možných aktivit, která infortatické myšlení rozvíjí a zároveň dovoluje samostatnou práci žáků, je programování. Hlavně programování robotů dovoluje žákům ihned vidět výsledek svojí práce tím, jak se naprogramovaný robot chová.

V rámci projektu PRIM vzniklo několik učebnic, včetně učebnice pro micro:bit. Ta je určena žákům středních škol a je vytvořena pro prostředí MicroPython. Záměrem této práce bylo tuto učebnici upravit a předělat tak, aby ji bylo vhodné použít na druhém stupni ZŠ. Mezi tyto úpravy patřilo přetvoření úloh z Pythonu do blokového prostředí MakeCode, přidání nových úloh a přepsání textů, které se do učebnice už nadále nehodily, jako jsou například texty zabývající se popisem MicroPythonu.

1.1 Cíl práce

Hlavním cílem práce je vytvořit učebnici programování programovatelné desky BBC micro:bit v blokovém prostředí MakeCode pro druhý stupeň ZŠ. Vychází z učebnice robotiky pro střední školy z projektu PRIM. Originální učebnice je určena pro žáky středních škol a zabývá se programováním micro:bitu v Pythonu.

Učebnice je rozdělena do několika kapitol. Každá kapitola se zabývá jednou funkcí micro:bitu, například prací s tlačítky, zvukem nebo vzájemnou komunikací dvou a více micro:bitů. Pomocí těchto funkcí si žáci vyzkouší práci s programy a jejich tvorbu. Učebnice je volně dostupná na internetu pod stejnou licencí jako originální učebnice.

Druhotným cílem je popsání různých blokových prostředí a jejich možností při programování micro:bitu. Kromě jiného jsou porovnány jejich možnosti při programování micro:bitu, jejich uživatelské rozhraní a přístupnost pro žáky

i učitele.

Posledním cílem práce, který byl stanoven až během vypracovávání, je navrhnout lepší překlad pro prostředí MakeCode, které v době vytváření učebnice neodpovídalo terminologii používané v jiných blokových prostředí a některé bloky nebyly přeloženy vůbec.

1.2 Metoda práce

Na začátku zpracovávání diplomové práce došlo k prostudování patřičné literatury. Hlavním zdrojem byla původní učebnice Robotika pro střední školy: programujeme micro:bit pomocí Pythonu. Kromě ní jsem hledal a studoval české i zahraniční publikace a webové portály zabývající se programovatelnou deskou micro:bit nebo infromatickým myšlením. Pro tvorbu úloh sloužil jako zdroj RVP pro základní školy, a to zejména klíčové kompetence a očekávané výstupy ve vzdělávací oblasti Informatika a informační a komunikační technologie. Nakonec byly nastudovány učebnice týkající se výuce programování v blokových prostředích.

Dále jsem zkoumal různá bloková prostředí a jejich možnosti při programování micro:bitu. Všechna prostředí jsem porovnal a otestoval. Následně jsem vybral Microsoft MakeCode jako nejvhodnější prostředí pro tuto práci.

Následovala analýza všech úloh v učebnici. Úlohy byly roztríděny do dvou kategorií na úlohy, které jdou a mají smysl být převedeny do blokového prostředí a na úlohy, které nejdou nebo nemají smysl být převedeny do blokového prostředí. Do této kategorie také mimo jiné patřily úlohy, které si za cíl braly jen izolovanou ukázkou některé z funkcí micro:bitu bez větší návaznosti na zbytek kapitoly. Také jsem do této kategorie zařadil úlohy, které neodpovídaly současnému RVP. První kategorie úloh pak byla předělána v prostředí Microsoft MakeCode a zahrnuta v učebnici. Úlohy z druhé kategorie byly nahrazeny vhodnějšími úlohami.

Původní učebnice je určena pro žáky středních škol, a proto jsou v ní ob-

čas dost velké skoky mezi obtížností jednotlivých úloh. Z toho důvodu byly přidány do určitých kapitol další úlohy, které obtížnost vyrovnají.

Zároveň s úlohami byly upraven i text učebnice, který se týkal popisu jazyka Python, na text k blokovému prostředí MakeCode.

Na základě připomínek autorů původní učebnice pak došlo k úplnému přepracování žakovských listů, které do té doby obsahovaly řešení úloh a texty s teorií. Také došlo k vyřazení verze učebnice určené k samostudiu.

Během předělávání úloh jsem došel ke zjištění, že názvy bloků česká mutace prostředí MakeCode kvalitou neodpovídají standardu jiných blokových prostředí, jako je například Scratch, a některé bloky ani nemají český název. Proto jsem se, společně s vedoucím této práce, rozhodl vypracovat návrh na kvalitnější český překlad názvů bloků.

Nakonec došlo ke zveřejnění učebnice na portálu iMyšlení. Učebnici lze zdarma stáhnout na následujícím odkazu: <https://tinyurl.com/UcebniceMakeCode>. Také byla přiložena k elektronické verzi této práce jako příloha a k fyzické verzi na CD.

2 Informatické myšlení

Informatické myšlení je termín, který v posledních několika letech nabývá důležitosti, v České Republice například při poslední revizi RVP. S nově nabytou popularitou ale přichází také několik pohledů a definic na tento pojem. Informatické myšlení hledá odpovědi na následující otázky: Co je to počítač? Čemu všemu musíme na počítači rozumět? Co všechno do něj musíme vložit, aby za nás vykonával naši práci? Jak počítače upravují náš pohled na svět? Co nového díky nim vidíme? Jakou roli hraje programování? Na jakou práci se počítače nehodí? [1]

Obecně řečeno je informatické myšlení schopnost myslet jako informatik při řešení problémů. Takto pojaté informatické myšlení má být užitečné i mimo informatiku, tedy pro každého.[2] Tato schopnost se projevuje v analýza a psání daného problému, a jeho co možná nejefektivnějším vyřešením.

Žáci s rozvinutým informatickým myšlením budou například umět:

- Systematicky posoudit různá řešení, vybrat to nejvhodnější pro danou situaci,
- rozdělit velký problém na několik menších, snáze řešitelných,
- plánovat a řídit činnosti,
- vytvářet a pečlivě popisovat postupy, které spolehlivě vedou k nějakému cíli, i když je vykonává někdo jiný,
- vybírat, které aspekty problému jsou podstatné pro řešení a které lze zanedbat,
- uspořádat i velké a nesourodé soubory dat tak, abychom je mohli dále využít
- používat jazyky, kterými se domluvíme s počítači, roboty a umělou inteligencí [3]

Rozvoj infromatického myšlení lze shrnout do několika bodů, podle kterých žákův proces myšlení probíhá:

1. Myšlenkový proces
2. Abstrakce
3. Dekompozice
4. Návrh algoritmu
5. Vyhodnocení
6. Generalizace
7. Automatizace[4]

Úlohy ve vytvořené učebnici jsou koncipovány tak, aby infromatické myšlení podle těchto bodů rozvíjely. Úlohy jsou stavěny na objevování nových možností Micro:bitu vlastní činností pomocí principu pokus-omyl nebo spoluprací ve skupince. Žákům je dána svoboda při návrhu vlastních řešení, a některé úlohy nemají jedno jediné řešení. Žáci mají možnost zapojit svého kreativního ducha, mohou pracovat vlastním tempem a rozhodnout se, do jaké hloubky se jednotlivými úlohami budou zabývat. Většina úloh obsahuje i návrhy pro rychlejší a šikovnější žáky, které je mají motivovat a inspirovat k dalšímu rozvoji svých dovedností.

Příkladem takových úloh je série úloh v kapitole Síť zabývající se posíláním zprávy zakódované pomocí Morseovy abecedy. V 8. úloze této kapitoly mají žáci za úkol vytvořit základní verzi programu, který po stisknutí jednoho tlačítka vykreslí na displeji přijímacího micro:bitu tečku a po zmáčknutí druhého tlačítka vykreslí čárku. V hodině předtím se žáci naučili, jak na displeji zobrazit písmeno A, pokud bylo stisknuto tlačítko A a B, pokud bylo stisknuto tlačítko B. Žáci generalizují řešení úlohy z minulé hodiny i pro tuto hodinu. Zároveň dochází k abstrakci – žáci si vyberou tu část kódu, která je důležitá pro posílání zpráv mezi micro:bity.

V následujících dvou úlohách žáci toto řešení vyhodnotí a otestují. Buď sami nebo s pomocí učitele přijdou na to, že je problém nejen s rozeznáním, kdy přišel nový znak, ale kde končí každé slovo zprávy a i celý přenos. Navrhnou vlastní řešení na odstranění těchto nedostatků.

Ve stejné kapitole se nachází i aktivita na vytvoření anketního systému. Aktivita je to dlouhá, proto ji žáci ve spolupráci s učitelem rozdělí na několik menších a lépe řešitelných problémů. Úloha se dá například rozdělit na následující části – zahájení hlasování na všech micro:bitech, poslání hlasu z hlasovacího micro:bitu, počítání hlasů na centrálním micro:bitu a zobrazení výsledků.

8. úkol

Najděte na internetu tabulku s Morseovou abecedou.

Vytvořte programy pro *vysílač* a *přijímač*, pomocí kterého půjde posílat zprávy zašifrované v Morseově abecedě.

Program otestujte posláním jednoho zašifrovaného písmene.

9. úkol

Učitel vám rozdá papírky se zprávou, kterou bude mít za úkol *vysílač* poslat *přijímači*.

Pokud během posílání zprávy narazíte na problém, programy upravte tak, aby k němu nedocházelo.

10. úkol

Nyní zakódujte a pošlete jednoslovnou zprávu.

Pokud během posílání zprávy narazíte na problém, programy upravte tak, aby k němu nedocházelo.

Poté pošlete víceslovnou zprávu. Znovu opravte všechny problémy, na které narazíte.

Obrázek 1: Žákovské zadání ukázkových úloh

3 Programovací deska BBC micro:bit

3.1 Obecné informace

Současná revize RVP dává možnost učitelům během výuky využít nové metody, postupy a pomůcky. Micro:bit je programovací deska, která umožňuje tyto nové metody a postupy jednoduše aplikovat ve výuce.

Micro:bit je open-source vývojový kit vyvinutý ve Velké Británii za podpory BBC, určený převážně pro výuku informačních technologií. [5] Byl navržen tak, aby byl vizuálně atraktivní, dával okamžitou zpětnou vazbu a byl cenově a uživatelsky dostupný.[6]

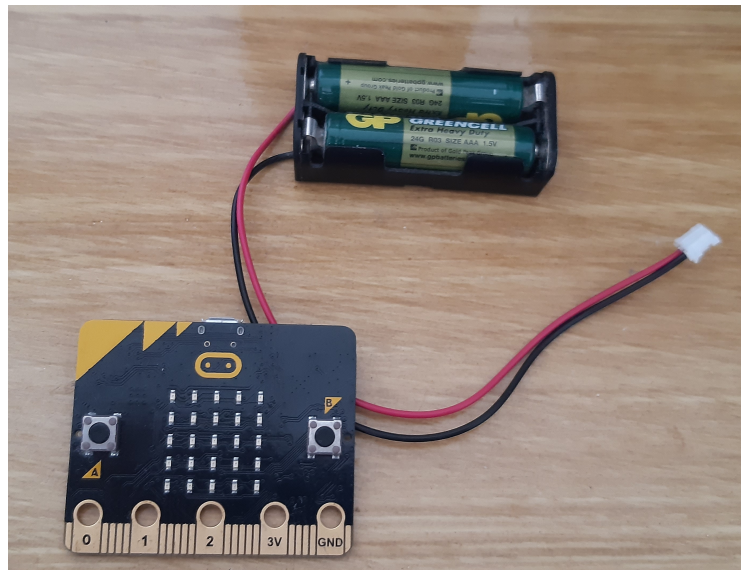
Micro:bit obsahuje displej tvořený 25 LED diodami, dvěma programovatelnými tlačítky, kompasem, 17 GPIO piny, gyroskopem, senzory teploty, světla a magnetického pole. Lze ho propojit s počítačem pomocí USB kabelu nebo přes Bluetooth. Dva Micro:bity spolu mohou komunikovat pomocí zabudovaného rádia.

V roce 2020 byla představena nová verze Micro:bitu, která se od původního Micro:bitu liší v tom, že navíc obsahuje dotykové tlačítko, zabudovaný mikrofon a silnější baterii a procesor. [7]

Micro:bit je možné programovat v mnoha různých prostředích, které se liší jednak obtížností na uživatele, ale i použitým programovacím jazykem. Tato práce se bude dále zabývat blokovými prostředími Microsoft MakeCode, Scratch, MicroBlocks a Vittascience, a okrajově se dotkne i prostředí Mu Python Editor, se kterým pracuje originální učebnice.

3.2 Historie

Micro:bit byl vytvořen ve Velké Británii. Světu byl představen 15. března 2015 při zahájení kampaně BBC jménem Make It Digital. [8] Ta si kladla za cíl inspirovat mladé angličany a vedla je k větší digitální gramotnosti. Micro:bit měl pomoci transformovat novou generaci z pasivních konzumentů technologií



Obrázek 2: Programovatelná deska BBC micro:bit

v jejich tvůrce a inovátory. [9]

V roce 2016 byl do anglických škol distribuován víc jak milion Micro:bitů. [10] Brzy se rozšířil i do zbytek světa. Původně byl určen pro žáky 2. stupně základních škol, ale uplatnění našel i na mnoha technicky zaměřených středních školách.

V roce 2020 byl vydán Micro:bit V2, který oproti první verzi obsahuje zabudovaný mikrofón, dotykové tlačítko, lepší baterii a silnější procesor.

3.3 Bloková prostředí pro programování Micro:bitu

V této kapitole budou porovnány různá bloková prostředí pro programování micro:bitu. Bude zohledněna jejich přístupnost a vhodnost pro programování micro:bitu, jejich využitelnost ve výuce a jejich obecné výhody a nevýhody, které by mohly hrát během vyučovací hodiny roli.

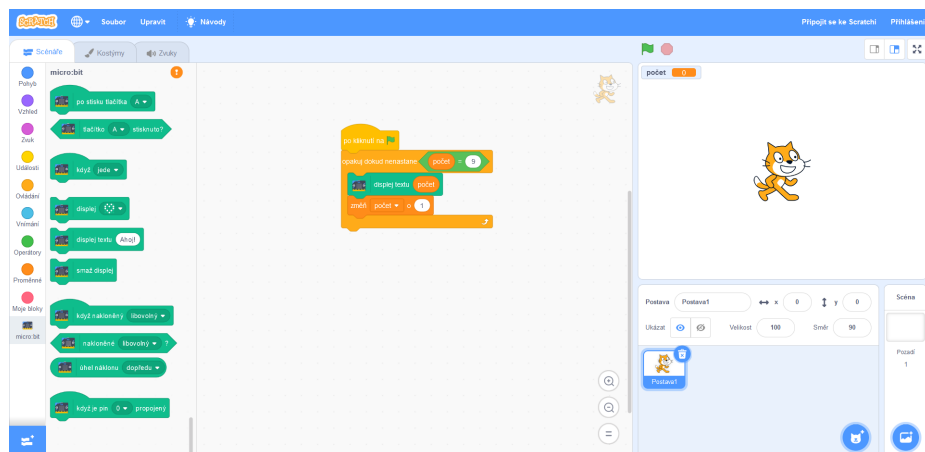
Jak už bylo řečený výše, zaměřím se na prostředí Microsoft MakeCode, Scratch, MicroBlocks a Vittascience.

3.3.1 Scratch

Scratch je populární volba pro výuku programování na základních školách.[11] Je plně v češtině a děti dostávají od prostředí okamžitou zpětnou vazbu k jejich kódu. Ta dovoluje žákům pracovat více samostatně.

Bohužel je propojení s micro:bitem pracné. Vyžaduje speciální program Scratch Link nainstalovaný na počítači. Také se do micro:bitu musí nahrát speciální program na propojení se Scratchem. [12]

Samotné prostředí pak poskytuje jenom základní nabídku bloků pro práci s micro:bitem, jako je reakce na stisknutí tlačítka nebo práce s displejem. Velké množství bloků ze Scratche, které žáci znají, nejsou s micro:bitem kompatibilní. Dají se ale různě kombinovat, například při naprogramování micro:bitu jako ovladače postavy ve Scratchi.



Obrázek 3: Prostředí Scratch

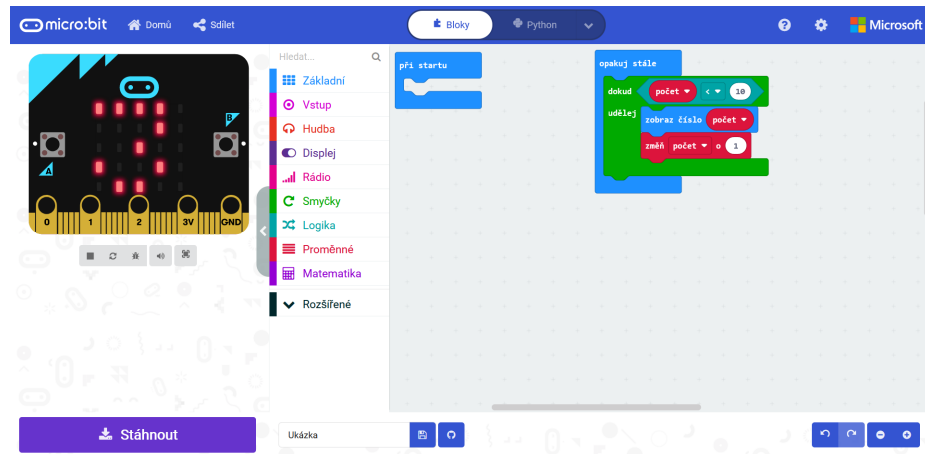
3.3.2 Microsoft MakeCode

Microsoft MakeCode je oficiální prostředí pro blokové programování micro:bitu. Součástí prostředí je digitální simulace micro:bitu. Ta dává žákům okamžitou zpětnou vazbu i bez fyzického micro:bitu. Prostředí je v češtině.

Všechny bloky souvisí přímo s micro:bitem a prostředí jich obsahuje nezměrné množství. Všechny bloky jsou úhledně poskládány do kategorií podle

funkce micro:bitu, se kterou pracují.

Kromě toho obsahuje MakeCode i možnost přepnutí do jazyka Python nebo JavaScript. Tato funkce se dá využít pro pokročilejší žáky.



Obrázek 4: Prostředí Microsoft MakeCode

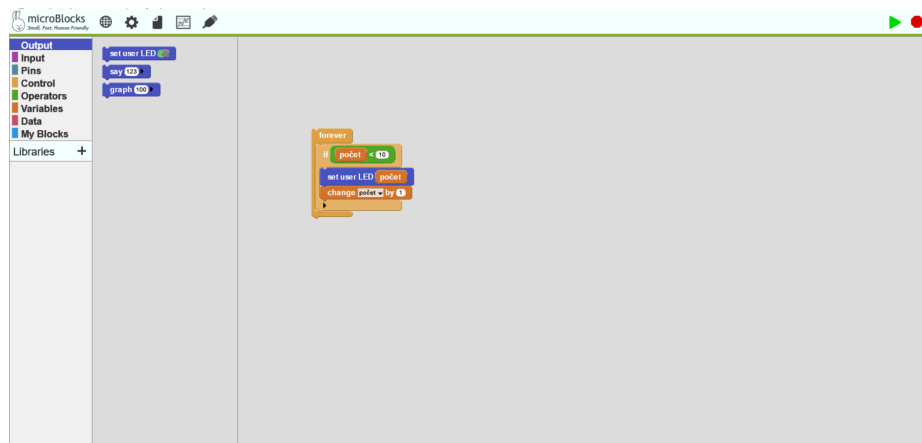
3.3.3 MicroBlocks

MicroBlocks je relativně nové prostředí. Je uvedeno jako jedno z doporučených prostředí na oficiálních stránkách micro:bitu. V současné době se nachází v betaverzi. [13]

Prostředí plně podporuje prohlížeče Google Chrome a Microsoft Edge. Během mého testování v omezené míře fungovalo i v prohlížeči Mozilla FireFox. Lze si také stáhnout desktopovou verzi. Prostředí je kompletně v angličtině a vyžaduje neustále připojený micro:bit.

Nabídka bloků je zatím velice chudá. Prostředí obsahuje minimum bloků pro práci s displejem a neobsahuje žádné bloky pro práci s dalšími funkcemi micro:bitu. Skládání bloků je neintuitivní, a na první pohled je těžké poznat, které bloky spolu jdou spojit.

Veliká výhoda je v obsáhlé knihovně bloků pro práci s různými rozšířeními a senzory. Také lze kliknutím spustit jednotlivé bloky kódů samostatně podobně jako ve Scratchi.



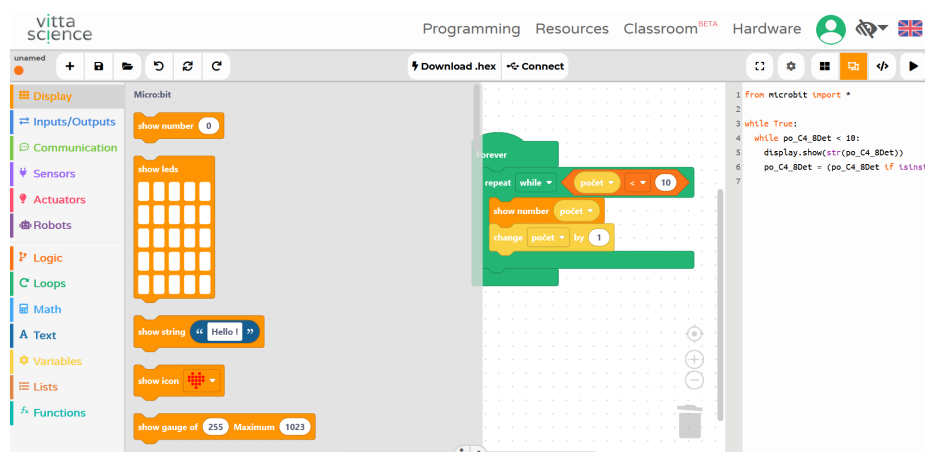
Obrázek 5: Prostředí MicroBlocks

3.3.4 Vittascience

Vittascience je další prostředí doporučováno na oficiálních stránkách micro:bitu.

Vittascience vyžaduje připojení k internetu. Je celé v angličtině. Obsahuje simulaci micro:bitu a lze v něm přepínat mezi bloky a Pythonem, lze ho tedy využít u starších a šikovnějších žáků k ukázce jazyka Python.

Bloky jsou řazeny do kategorií a pak dále rozděleny podle použití. Nabídka bloků je podobná té z prostředí MakeCode, ale obsahuje i bloky určené pro nejpoužívanější rozšíření Micro:bitu jako je například Neo Pixel.



Obrázek 6: Prostředí Vittascience

3.3.5 Shrnutí

Ve všech prostředích došlo k otestování stejné sada úloh týkající se různých funkcí Micro:bitu. Úlohy se týkaly prací s displejem, zvukem, rádiem, kompasem a připojením teplotního čidla. Jednalo se tedy o podobnou sadu úloh, jako kterou bude nabízet hotová učebnice.

Úlohy zaměřené na displej se zabývaly jednoduchostí vykreslování, vzhledem k tomu, že by se mělo jednat o první funkci, se kterou se žáci seznámí. Byla preferována možnost, kdy si žáci rozsvícení LED diod naklikají v matici 5x5 symbolizující displej micro:bitu, ale byla vítaná i možnost ovládat jednotlivé LEDky pomocí souřadnic.

Úlohy se zvukem byly v podstatě převzaty z původní učebnice a bylo otestováno jejich možné předělání v blokovém prostředí. Mezi těmito úlohami se nacházely hlavně takové úlohy, ve kterých si žáci vytvářeli vlastní melodie. Bylo preferováno grafické prostředí jako například virtuální piáno namísto prostého psaní tónů, protože ne všichni žáci mohou notám rozumět.

Úlohy na vzájemnou komunikaci micro:bitů se zaměřovaly převážně na bezdrátovou komunikaci pomocí rádia, protože žádné z prostředí neposkytovalo jednoduchou možnost jak propojit micro:bity kabely.

MicroBlocks měl problém s rozsvěcováním LEDek displeje, protože neobsahuje žádný blok pro specifikování LEDek k rozsvícení. MicroBlocks a Scratch nemají žádné možnosti, jak pracovat s funkcemi jako je kompas nebo teploměr. Také v nich není nijak možné vytvořit programy na komunikaci dvou nebo více micro:bitů.

V MakeCodu a Vittascience se mi povedlo úspěšně vyřešit všechny úlohy. Dále jsem se rozhodl úlohy do učebnice vytvářet v MakeCodu. Vittascience působil místy nepřehledně, kvůli množství různých bloků. V kategorii pro vstupy a výstupy jsou tak například obsaženy i bloky pro práci s mikrofonom a piny. Pro oboje existují v MakeCodu vlastní kategorie, čímž je orientace v prostředí jednodušší. Také je na obrazovce najednou mnoho oken, které se musí při kaž-

dém spuštění pozavírat, jako je konzole nebo Python editor. Nevýhodou také byla absence českého překladu.

Pro tvorbu úloh do učebnice byl tedy MakeCode nejlepší. Prostředí Vitasience bych doporučil využít na kroužcích robotiky při práci s příslušenstvím, jako jsou vozítka, LED pásy nebo různé senzory. Také by mohl být vhodnou variantou pro žáky středních škol, kteří zvládnou potřebnou úroveň angličtiny. MicroBlocks a Scratch jsem vyhodnotil jako nevhodné pro práci s micro:bitem, kvůli absenci bloků pro práci se základními funkcemi micro:bitu.

Úlohy do učebnice jsem se proto rozhodl vytvářet v MakeCodu. Český překlad tohoto prostředí ale kvalitou neodpovídal standardu dalších prostředí. Proto jsem se rozhodl před vytvářením úloh celé přepracovat.

3.4 Porovnání Microsoft MakeCode a Mu Python Editoru

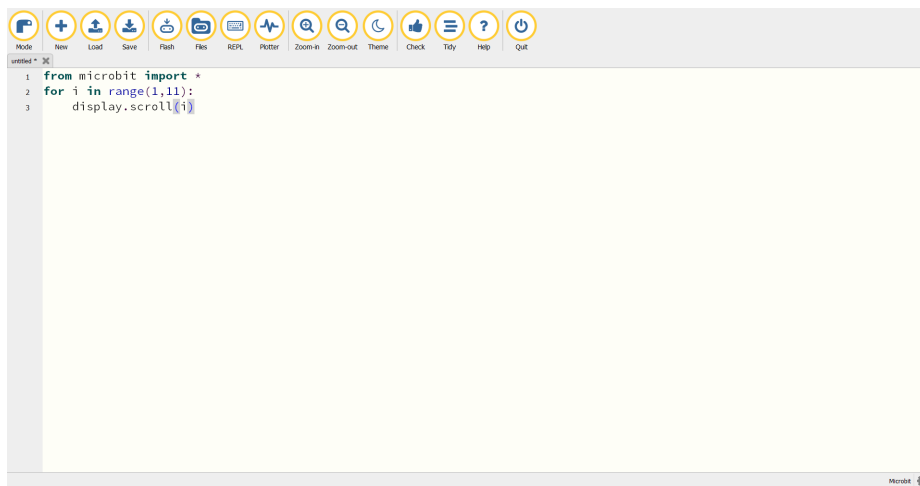
Původní učebnice Robotika pro střední školy: programujeme Micro:bit pomocí Pythonu má úlohy vytvářené v Pythonu a prostředí Mu Python Editor. Na rozdíl od blokových prostředí dávají uživateli více prostoru k chybám, které se netýkají logické správnosti kódu.

Oproti blokovému prostředí, jako je MakeCode, hrozí překlepy v příkazech kódu. Žáci tak mohou napsat logicky správný kód, ale udělat chybu v jednom z příkazů. Například mohou zapomenout nebo přidat do příkazu písmeno, u proměnných zapomenout nebo změnit velikost jednotlivých písmen nebo napsat špatný počet závorek u příkazů. Jednotlivé příkazy si navíc žáci musí pamatovat nebo najít v dokumentaci, jinak s nimi nemohou pracovat. V neposlední řadě je jazyk MicroPython v angličtině, žáci tedy nemusí všem příkazům hned rozumět nebo si je vybavit. Mu Python Editor také striktně dodržuje odsazení kódu. Při jeho porušení se program nepřeloží.

Microsoft MakeCode všechny problémy popsané v předchozím odstavci odstraňuje. Žáci nemusí kódy psát, čímž odpadá nutnost kontroly gramatiky.

Nemusí si jednotlivé příkazy pamatovat, protože je jednoduše najdou v nabídce bloků. To také dovoluje žákům samostatně hledat a zkoumat nové bloky a učit se je použít. Všechny bloky jsou v češtině a barevně a tvarově rozděleny podle funkce, kterou vykonávají, žáci tedy mohou intuitivně odhadnout, k čemu každý blok slouží. Při práci s proměnnými mají vždy k dispozici seznam všech proměnných, takže se nemůže stát, že by v jejich názvech udělali chybu. Prostředí žákům při spojování kódů ukazuje, kam nový blok zapadne. Lze tak už na první pohled vidět, jestli je blok umístěn v kódu správně.

Blokové prostředí MakeCode je vhodnější pro žáky 2. stupně ZŠ. Na rozdíl od Mu Python Editoru umožňuje efektivní samostatnou práci nebo práci v malých skupinkách. Nadanějším žákům navíc dovoluje se do MicroPythonu přepnout a sledovat, jak se psaný kód mění s každým přidaným a odstraněným blokem. Moje upravená učebnice předpokládá a doporučuje, aby se žáci s micro:bitem seznámili až poté, co budou umět programovat a pracovat s bloky ve prostředí Scratch. I z toho důvodu pro ně bude MakeCode vhodnější než Mu Python Editor.



Obrázek 7: Prostředí Mu Python Editor

4 Překlad prostředí Microsoft MakeCode

Během vytváření a testování úloh jsem došel k zjištění, že česká mutace prostředí Microsoft MakeCode nedosahuje dostatečné kvality pro použití ve výuce.

Nacházely se zde nepřeložené nebo strojově přeložené bloky. Terminologie uvnitř bloků se často lišila. Například v blocích pracujících se seznamy byl termín *seznam* (v anglické verzi označován jako *array*, anglické označení pro pole) přeložen několikrát jako *seznam*, ale také jako *list*. V původní anglické verzi se nazývaly struktury, správně zvané seznamy, jako pole. Na některých místech bylo dokonce ponecháno původní označení *array*. V takových případech jsem vždy volil to označení, které se pro stejný termín nachází v prostředí Scratch. Pokud tam neexistoval ekvivalentní termín, poradil jsem se s vedoucím práce, který na překlad dohlížel, a použil termín, který doporučoval on.

Bloky se lišily i s tvarem slovesa. Velká většina bloků používala infinitiv sloves, ale občas se objevil blok, který z této normy vybočoval a obsahoval jiný tvar slovesa.

Zde jsem se znovu rozhodl přistoupit ke slovesům stejným způsobem jako Scratch. Všechny jsem přepsal do 2. osoby jednotného čísla rozkazovacího způsobu.

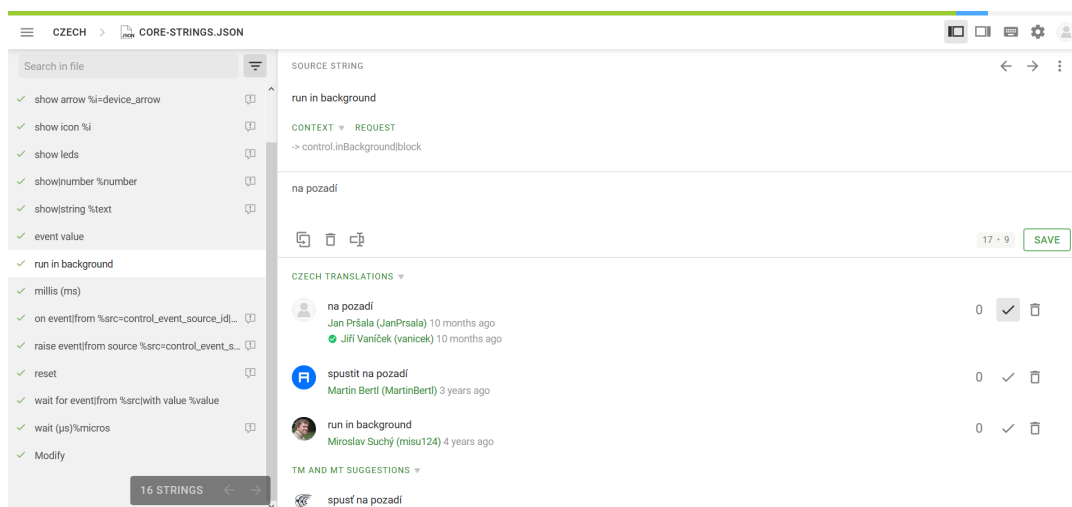
I přes moje snahy se pořád v prostředí nachází několik anglických výrazů. Například proměnné v sekci pro práci s rádiem jako *receivedNumber* nebo *receivedString*. Souborů pro překlad bylo několik a některé byly společné pro micro:bit, Minecraft: Education Edition a další zařízení. K souborům s termíny, které jsou společné pro více zařízení, jako jsou zmíněné proměnné, jsem bohužel neměl ani přístup.

U jednotlivých textů nešlo na první pohled poznat, jestli obsahují název bloku, jeho popis nebo jinou část prostředí. V souborech také nešlo dobře vyhledávat. Některé texty proměnných se mi povedlo najít mimo hlavní soubory pro micro:bit, ale ne úplně všechny. Z toho důvodu jsem byl nucen zbytek proměnných nechat nepřeložený.

Podobný problém byl i u výchozích textů, do kterých může uživatel napsat vlastní hodnoty. Například blok pro zobrazení textu na displeji `micro:bitu` má jako výchozí text výraz *Hello!*. Tyto texty se v souborech nenacházejí a jsou řešeny na úrovni, ke které jsem neměl přístup. Naštěstí se takových textů v blocích moc nevyskytuje, a pokud ano, jedná se o jednoslovné známé anglické fráze.

Překlad probíhal na stránkách Crowdin. Toto prostředí práci na překladu nezjednodušilo. Bylo těžké se v něm orientovat a vyhledat přesně ty soubory, které souvisí s `micro:bitem`. Samotné soubory pak obsahovaly jak názvy bloků a proměnných, tak i jejich popisky a další texty, které s prostředím souvisí, jako je například překlad nastavení. Nebylo ale nikde uvedeno, k jakému bloky který popisek patří, což u některých špatných překladů vytvořilo další problémy při jejich opravách.

Překlad tedy proběhl bez větších problémů. Všechny dostupné bloky byly přeloženy. Stejně tak i proměnné, ke kterým jsem měl přístup. Jediná anglická proměnná, se kterou se v učebnici pracuje je proměnná `receivedNumber` v kapitole Síť. Po dokončení překladu a dopsání učebnice bylo prostředí aktualizované a byly přidány bloky pro práci s `micro:bitem` verze 2. Tyto bloky nebyly v době psaní této práce přeloženy.



Obrázek 8: Ukázka prostředí Crowdin

5 Přehled úloh v původní učebnici

Úlohy v původní učebnici Robotika pro střední školy: programujeme Micro:bit pomocí Pythonu jsou rozděleny do 7 kapitol. Každá z nich se zaměřuje na jednu funkci micro:bitu. Kapitoly jsou dále děleny podle jednotlivých hodin, a následně pak do kroků, u kterých je i odhadnuta časová náročnost. Úlohy v učebnici jsou stavěny především na opisování kódu a jeho případných úpravách.

Všechny úlohy jsem rozdělil do dvou skupin. Na úlohy, které jdou a mají smysl být převedeny do blokového prostředí a na úlohy, které nejdu nebo nemají smysl být převedeny do blokového prostředí.

5.1 Kapitola 1 - LED displej

První kapitola se zaměřuje na ovládání a práci s LED displejem a zároveň slouží jako seznámení s micro:bitem. Kapitola je připravena až na 5 vyučujících hodin.

5.1.1 Hodina 1

První dva kroky z první hodiny slouží k představení micro:bitu žákům. Micro:bit je v nich podrobně popsán.

Třetí krok popisuje fungování Mu Python Editoru. Tento krok bude nahrazen podobným odstavcem o editoru Microsoft MakeCode.

Poslední krok hodiny obsahuje ukázkovou úlohu, ve které žáci nechají po micro:bitu jednou proběhnout text "Ahoj svete". Tato úloha je začátek vhodná, bude proto zahrnuta v nové verzi učebnice. Místo textu "Ahoj svete" bude pro jednoduchost použit jen text "Ahoj".

5.1.2 Hodina 2

První krok druhé hodiny navazuje na předcházející úlohu. Tentokrát je žákům předložen program, který bude předchozí text psát pořád dokola. I tato úloha

bude zahrnuta v nové učebnici.

V dalších dvou úlohách z této hodiny jsou žákům představeny cykly *for* a *while*. Žákům je poskytnut program pro vypsání čísel od 1 do 10. I tyto úlohy budou převedeny do nové učebnice, ale bude upraven interval čísel. Protože číslo 10 na micro:bitu "protéká" po displeji, místo toho, aby se celé zobrazilo, jako se děje v případě jednomístných číslic. Proto budou žáci pracovat s intervalem od 0 do 9.

Poslední krok této hodiny obsahuje úlohu, ve které se žáci seznámí s příkazem MicroPythonu *display.show*. Tento příkaz nemá v MakeCodu ekvivalentní náhradu a jedná se o izolovanou ukázkou MicroPythonu, proto nebude zahrnuta v nové učebnici.

Ukázka dalších funkcí pro objekt *display*:

```
from microbit import *
display.show("X")
sleep(1000)
display.clear()
```

Příklad zobrazí znak X pomocí *display.show()* po dobu jedné sekundy a pak smaže displej pomocí *display.clear()*. Funkce *display.show()* zobrazí řetězec nebo číslo znak po znaku. Prodlevu mezi změnami lze nastavit (v milisekundách pomocí parametru. Např. *display.show("Ahoj svete", 1000)*. Poslední znak zůstane svítit na displeji. Pokud je parametrem jen jeden znak nebo jednociferné číslo, pak toto zůstane svítit tak dlouho než zobrazíte něco jiného nebo smažete obrazovku pomocí *display.clear()*.

Obrázek 9: Ukázka nevyhovující úlohy v původní učebnici

5.1.3 Hodina 3

Třetí hodina se zaměřuje na vykreslování obrázků na displeji. První úloha ukazuje práci s předpřipravenými obrázky. Tato úloha bude zahrnuta.

Další úloha pomocí dvou předpřipravených obrázků simuluje bijící srdce. Tato úloha měla být původně v učebnici zahrnuta, ale později byla odstraněna a nahrazena jinými úlohami.

Poslední úloha žákům představila možnost tvorby vlastních obrázků, včetně jasu jednotlivých diod. MakeCode obsahuje blok, pomocí kterého mohou žáci

generovat vlastní obrázky, ale na rozdíl od MicroPythonu neumožňuje nastavit jas diod. Tato úloha nebude zahrnuta v učebnici.

Na konci této kapitoly je poskytnut odkaz na generátor obrázků. Vzhledem k tomu, že tato funkce je v MakeCodu již zabudovaná, bude tento odkaz smazán.

5.1.4 Hodina 4-5

Poslední hodina této kapitoly se zabývá tvorbou animací poskládáním několika obrázků za sebou. Vzhledem ke svojí délce je možné tuto hodinu rozdělit na dvě.

První úloha navazuje na předchozí úlohu z minulé hodiny. Tato úloha je vhodně postavená pro ukázkou animací, proto bude zahrnuta s malou úpravou. Vzhledem k tomu, že MakeCode neumožňuje na této úrovni upravovat jas, budou všechny diody rozsvícené na maximální úroveň. V úloze je také ukázán seznam, který ale pro funkčnost programu v MakeCodu není potřeba, proto nebude použit.

Další úloha se zabývá prací jen s jednou diodou pomocí souřadnic. Úloha má dva programy. První program v úloze vybere náhodně jednu z 25 diod a rozsvítí ji náhodně silným jasnem. Druhý program dělá to samé, ale bez náhodného jasu. Programy mají symbolizovat analogové a digitální zobrazení. Tyto úlohy bohužel nelze v MakeCodu jednoduše sestavit, proto nebudou zahrnuty. Místo nich bude zahrnuto několik kratších a jednodušších úloh na práci s konkrétní diodou.

5.2 Kapitola 2 - Tlačítka

Tato kapitola představuje práci s tlačítky. Časová dotace této kapitoly je 1 vyučující hodina.

5.2.1 Hodina 1

První úloha obsahuje ukázkou práce s tlačítky A a B na micro:bitu. Pomocí větvení programu a podmínek je vytvořen program, který po zmáčknutí tlačítka A vykreslí usmívající se obličej a po stisknutí tlačítka B vykreslí smutný obličej. MakeCode obsahuje speciální bloky pro kontrolu stisku těchto tlačítek, proto je převedení této úlohy do blokového prostředí jednoduché. Tato úloha bude zahrnuta v učebnici.

Další úloha pak na stisknutí tlačítek ukazuje fungování logických spojek *and* a *or*. Logická spojka *and* je pro tlačítka předpřipravena i v MakeCodu, proto se tato část úlohy může zahrnout v učebnici. Pro logickou spojku *or* to ale neplatí.

Poslední úloha se zabývá funkcí MicroPythonu *get_presses()*, pro kterou neexistuje ekvivalentní blok, ale jeho funkce se dá lehce v MakeCodu sestavit. Lze tedy tuto úlohu zahrnout v učebnici.

Na konci kapitoly se nachází neřešená úloha, ve které mají žáci za úkol naprogramovat postřehovou hru. Tuto úlohu lze zahrnout v učebnici, vzhledem k tomu, že všechny požadavky, které úloha má, je možné naprogramovat i v MakeCodu.

5.3 Kapitola 3 - Zvuk

Tato kapitola se zabývá prací se zvukem. Časová dotace kapitoly jsou 2 hodiny.

5.3.1 Hodina 1

V prvním kroku hodiny je popsán způsob propojení micro:bitu a reproduktoru. Tato část je srozumitelně a podrobně sepsaná, proto bude využita v učebnici.

První úloha ukazuje základní možnost pro přehrávání zvuků. Je to vhodný úvod do kapitoly, proto bude úloha ponechána v učebnici.

Druhá úloha kombinuje vše, co se zatím žáci naučili. Na stisknutí tlačítka se rozsvítí displej a přehraje zvuk. I tato úlohu lze převést do nové učebnice.

5.3.2 Hodina 2

První úloha druhé hodiny předvádí funkci MicroPythonu `speech.say()`. Pro tu neexistuje v MakeCodu ekvivalentní blok, proto ji nelze použít v nové učebnici.

V poslední úloze mají žáci za úkol sestavit vlastní melodii. Blok pro vytváření vlastních melodií se v MakeCodu nachází, navíc tato úloha dovoluje žákům vymýšlet vlastní kreativní řešení, proto bude zahrnuta v učebnici.

5.4 Kapitola 4 - Poloha

Tato kapitola se zabývá prací s polohou, gesty a vstupy, které micro:bit dokáže rozpoznat. Časová dotace kapitoly jsou 4 vyučovací hodiny.

5.4.1 Hodina 1

Před zadáním první úlohy je popsáno, jak se micro:bit chová při naklonění podle os x, y, z. Samotná úloha pak slouží jako praktická ukázka této teorie. Tuto úlohu lze částečně převést do blokového prostředí. MakeCode dovoluje sledovat jen změnu osy x.

Další úloha představuje žákům kód simulátoru hudebního nástroje Thereminu. K jeho fungování je bohužel potřeba sledovat změnu na ose y, proto tuto úlohu nelze převést do blokového prostředí.

5.4.2 Hodina 2

Druhá hodina se zabývá gesty. První úloha slouží k seznámení s tímto pojmem. Pomocí gest jako je naklání nebo zatřesení se různě rozsvítí diody na displeji. Tuto úlohu lze použít v nové učebnici.

Druhá úloha simuluje magickou kouli. Po zatřesení se na displeji vypíše jedna z předpřipravených zpráv vybraných ze seznamu. Úlohu lze převést do nové učebnice.

5.4.3 Hodina 3

Třetí hodina popisuje práci s kompasem. První úloha ukazuje, jak kompas zkalibrovat, a co se vypisuje na displeji. Tuto úlohu je možné realizovat i v MakeCodu.

Následující úloha pak vytváří simulaci kompasu, který na displeji ukazuje na jakou světovou stranu je natočen. Úloha je těžká, ale pořád realizovatelná i v blokovém prostředí.

Poslední úloha této hodiny pracuje s polem obrázků MicroPythonu jménem *Image.ALL_CLOCKS*. Žákům je předán program, který právě díky tomuto poli ukazuje vždy na sever. Úlohu nelze převést do MakeCodu bez velkých úprav.

5.4.4 Hodina 4

Čtvrtá hodina této kapitoly se zabývá měřením magnetického pole. Jediná úloha v této hodině je obtížná a plná složitých výpočtů, ale lze ji převést do blokového prostředí. Proto bude v učebnici obsažena, ale jen jako dobrovolná úloha.

5.5 Kapitola 5 - Síť

Tato kapitola se zabývá propojením a komunikací dvou a více micro:bitů. Časová dotace jsou 4 vyučovací hodiny.

5.5.1 Hodina 1

Na začátku hodiny je znova uvedena teorie k probíranému tématu. Tato teorie bude zachována.

Jediná úloha v této hodině se zabývá propojením a komunikací dvou micro:bitů kabely. Bohužel toto v MakeCodu nelze realizovat, proto tato úloha nebude zahrnuta v nové učebnici.

5.5.2 Hodina 2

Tato hodina se zabývá sériovým přenosem. Bohužel znovu požaduje propojení micro:bitů kabelem, nelze tedy žádnou úlohu z této hodiny převést do MakeCodu.

5.5.3 Hodina 3

Tato hodina se zabývá obousměrným přenosem. Bohužel znovu požaduje propojení micro:bitů kabelem, nelze tedy žádnou úlohu z této hodiny převést do MakeCodu.

5.5.4 Hodina 4

Tato hodina se zabývá rádiovým přenosem. K němu je v MakeCodu vytvořena odpovídající sekce, lze tedy ukázkovou úlohu tohoto přenosu zahrnout v nové učebnici.

5.6 Kapitola 6 - Periférie

Tato kapitola popisuje propojení micro:bitu s různými externími perifériemi. Například s teplotním čidlem nebo světelnou diodou. Časová dotace této kapitoly jsou 3 vyučovací hodiny.

5.6.1 Hodina 1

V první hodině je předvedeno propojení trojbarevné diody s micro:bitem. První úloha slouží na ukázkou základních funkcí diody. Druhá úloha pak žáky učí, jak na diodě postupně rozsvítit a zhasnout jednu barvu. Obě úlohy lze převést do MakeCodu a budou zahrnuty v učebnici.

5.6.2 Hodina 2

Druhá hodina obsahuje jen jednu úlohu, která navazuje na předchozí úlohu. Žáci v ní mají za úkol vytvořit lampu, která náhodně střídá barvy, kterými

svítí. Program prezentovaný v úloze je možné poskládat v MakeCodu. Jedná se o zajímavou aplikaci micro:bitu, proto bude v nové učebnici zahrnuta.

5.6.3 Hodina 3

Poslední hodina se zabývá propojením micro:bitu a teplotního čidla. Program poskytnutý v jediné úloze kapitoly lze převést do MakeCodu a tato celá hodina bude zahrnuta v učebnici.

5.7 Závěrečný projekt

Učebnice obsahuje dva závěrečné projekty, mezi kterými si můžou žáci vybrat. Oba pracují s perifériemi, proto budou v nové učebnici zahrnuty ve odpovídající kapitole.

V prvním projektu se staví dálkově ovládané teplotní čidlo a ve druhém zabezpečovací zařízení. Oba projekty jsou stručně popsány. Jejich realizace je pak celá ponechána na žácích.

5.8 Žákovské listy

Výše popsané kapitoly byly brány z metodických listů určených pro učitele. Učebnice také obsahuje pracovní listy pro žáky. Ty jsou až na pár odebraných textů totožné s metodickými listy. Obsahují popsanou teorii a zadání i řešení všech úloh. Není proto nutné je zde podrobněji popisovat.

5.9 Samostudium

Sekce pro samostudium obsahuje podrobně popsanou teorii a sbírku všech úloh z hodin, včetně jejich řešení. Úlohy budou přepracovány podle výše napsaného a celá tato sekce bude zahrnuta v nové učebnici.

6 První verze nové učebnice

Po analýze a otestování původní učebnice jsem začal pracovat na vlastní podobě učebnice. Tato verze vznikala od léta do podzimu roku 2020 a prošla několika proměnami. Na následujících stránkách bude popsána finální podoba učebnice, kterou jsem dále posílal původním autorům pro získání jejich názoru a zpětné vazby.

Struktura učebnice je stejná. Každá kapitola obsahuje tři verze. Metodiku pro učitele, pracovní listy pro žáky a teorii pro samostudium.

6.1 Obecné změny v učebnici

Vybrané úlohy z původní učebnice byly převedeny do blokového prostředí. Jejich zadání bylo až na případné zmínky MicroPythonu zanecháno. Kód napsaný v Mu editoru, který byl u každé úlohy poskytnut, byl nahrazen ekvivalentním kódem z MakeCodu. Stejně tak byly odstraněny i další texty týkající se MicroPythonu nebo Mu editoru. Teorie byla na několika místech upravená, aby více odpovídala předpokládaným znalostem žáků základních škol. Styl úloh, kdy byl žákům kód předložen k opsání byl u velké většiny úloh zanechán.

V učebnici jsou chybně žáci označováni jako studenti. Také občas není dodrženo označení pro různé speciální části textu, jako je otázka či informace. Tyto oba nedostatky jsem odstranil.

Tyto změny se děly u všech měněných úloh, nebudou tedy už zvlášť zmiňovány u jednotlivých kapitol.

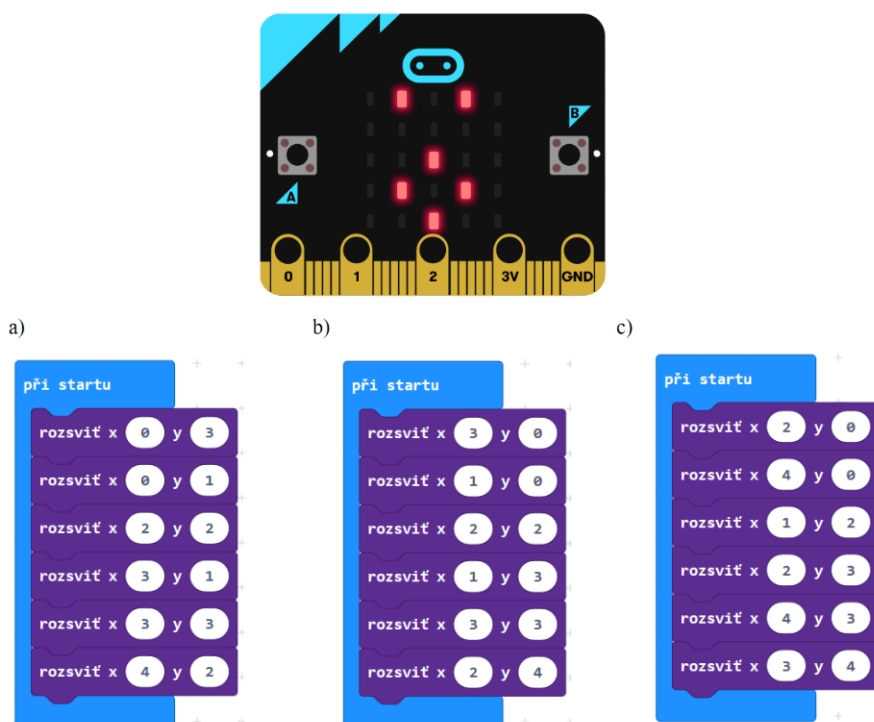
6.2 Nové úlohy

V původní učebnici na několika místech došlo k velkému skoku v obtížnosti úloh. Například po původní úloze o kompasu, kde se s touto funkcí žáci teprve učí zacházet, následovala těžká úloha naprogramování klasického kompasu.

Vzhledem k tomu, že z učebnice bylo odstraněno mnoho úloh, těchto roz-

dílů v obtížnosti a dalších prázdných míst se objevilo ještě víc. Proto jsem do každé kapitoly přidal vlastní úlohy. Tyto úlohy byly vytvářeny podle cílů a očekávaných výstupů nových RVP. Kódy v nich nejsou vždy žákům přímo prezentovány. Místo toho jsou žáci vyzváni k vytvoření vlastního kódu. Ke správnému řešení jsou naváděni pomocí různých otázek, na které hledají odpovědi. Úlohy se ne vždy zaměřovaly přímo na tvorbu kódů, ale i na jejich pochopení a čtení. Tím byla zajištěna větší rozmanitost úloh.

Nyní žákům ukažte vytvořený obrázek na micro:bitu a předložte jim následující scénáře. Vyzvěte je, ať vyberou správný scénář: (Řešení – b))



Obrázek 10: Ukázka přidání úlohy

6.3 Popis změn po kapitolách

V této sekci budou popsány všechny kapitoly a změny, které v nich oproti původní učebnici byly provedeny.

6.3.1 Kapitola 1

První a druhá hodina jsou téměř totožné s původní učebnicí. Jediný rozdíl je v intervalu vypisování čísel na displeji micro:bitu pomocí cyklů. MakeCode nepovoluje mít jinou dolní hranici než 0.

Ve třetí hodině této kapitoly byly přidány dvě úlohy. První úloha seznamuje žáky s tvorbou vlastních obrázků pomocí bloku *ukaz tvar*. Druhá úloha ukazuje, jak stejný obrázek na micro:bitu rozsvítit pomocí postupného rozsvěcování jednotlivých diod. Adresace diod byla v původní učebnici probírána až ve čtvrté hodině, ale tématicky se hodí více do této hodiny.

Ve čtvrté a páté hodině došlo k velkým změnám. První úloha s animací zůstala nezměněna, ale další úlohy jsou nové. Protože byl úvod k práci se specifickou diodou přesunut o hodinu zpět, věnuje se zbytek kapitoly práci právě s určitými diodami pomocí souřadnic. V první úloze je žákům předložen kód a oni mají bez použití micro:bitu odhadnout, co se na něm vykreslí. Následující dvě úlohy navazují přidáním dalších bloků do kódu, tentokrát bloky, které diody zhasnou. Další úloha žákům odhaluje kód smutného obličejce a vyzývá je, aby kód pomocí svých dosavadních znalostí upravili tak, aby se obličej usmíval. Upravovat kód mají žáci za úkol i v následující úloze. V něm jim je dán špatný kód a oni ho mají za úkol opravit. V poslední úloze žáci znovu micro:bit odloží, protože budou mít za úkol vybrat jeden ze tří kódů, který vykreslí na displeji ukázaný obrázek.

6.3.2 Kapitola 2

Ukázka práce s tlačítky A, B a jejich kombinace jsou s původní učebnicí totožné. Stejně tak i úloha na konci, ve které dochází k vytvoření programu počítající počet stisknutí tlačítka A. Přidané jsou úlohy s gesty. Gesta jsou podobně jako tlačítka druh podmíněného vstupu, proto mi přišlo logické je zde alespoň zmínit, i když se s nimi bude pracovat více až ve 4. kapitole. Tyto úlohy nahrazují úlohy s logickými spojkami *and* a *or* z původní učebnice.

Nyní žákům ukažte následující scénář, který vypíše čísla od 0 do 9. Řekněte jim, v jaké sekci příslušné bloky najít:



Vysvětlíte scénář. Jedná se o *cyklus s pevným počtem opakování*. Je třeba vysvětlit, že blok pořadí mění svoji hodnotu a lze ho použít jen uvnitř cyklu. Také je potřeba dát pozor na to, že rozsah od 0 do 9 v *MakeCode* znamená, že se scénář provede celkem 10x, tedy jedenkrát více, než je mez vpravo.

Obrázek 11: Ukázka upravené úlohy

6.3.3 Kapitola 3

První hodina je totožná s první hodinou v původní učebnici.

Úlohy ve druhé hodině jsou změněny. Na začátek je posunuta úloha, ve které si žáci vytváří vlastní melodie. Přidána je ukázka několika známých melodií jako je vánoční koleda *Rolničky* nebo první sloka písničky *Ovčáci čtveráci*, které jsou v *MakeCode* sestaveny. Nakonec je na místo odstraněné úlohy se čtením textu vložil úlohu, která pracuje s hudbou jako s podmínkou.

6.3.4 Kapitola 4

Vzhledem k tomu, že v první kapitole došlo k odstranění úlohy s Thereminem, rozhodl jsem se základní práci s gesty přesunout ze začátku druhé hodiny na konec této hodiny. Podobná úloha se už objevila ve 2. kapitole. Zde je pro připomenutí práce s gesty, a také proto, že se s gesty bude víc pracovat v příští hodině.

Druhá hodina začíná novou úlohou, která slouží jako náhrada za program s magickou koulí. Žáci mají za úkol sestavit program, který bude simulovat hru Kámen, nůžky, papír. Po zatřesení se na *micro:bitu* rozsvítí K pro Kámen, N pro nůžky nebo P pro papír. Žáci si tak mohou zahrát sami se sebou. Jako nápověda pro žáky slouží další úloha, ve které mají za úkol vytvořit program,

který po zmáčknutí tlačítka A na displeji ukáže náhodné číslo od 0 do 9. Je zde také navrhnutá aktivita pro žáky na vyzkoušení řadících algoritmů. Žáci se seřadí před tabulí a musí se seřadit za použití pravidel určitého řadícího algoritmu.

Ve třetí hodině přibyla úloha, ve které mají žáci za úkol vytvořit program, který vypíše V, pokud je micro:bit natočen více na východ, a Z, je-li micro:bit více natočen na západ. Také byl přidán obrázek ukazující intervaly stupňů pro všechny světové strany. Ten může žákům pomoci si fungování kompasu lépe představit.

Poslední hodina byla nezměněna.

6.3.5 Kapitola 5

Celá kapitola týkající se propojení více micro:bitů byla přepracována. Vzhledem k tomu, že v MakeCodu nelze dobře propojit dva micro:bity kabely, byla celá kapitola upravena tak, aby spolu micro:bity komunikovaly jen pomocí zabudovaného rádia. S úmyslem ponechání stejné časové dotace pak byly do kapitoly přidány úlohy týkající se praktické aplikace bezdrátové komunikace micro:bitů, jako je například kódování a šifrování.

V první hodině se žáci dozví základní informace o sítích a zkusí si ve dvojicích propojit dva micro:bity. Jeden micro:bit bude sloužit jako vysílač a druhý jako přijímač signálu.

V druhé hodině mají žáci nejdříve za úkol odhalit, co je uloženo v proměnné *receivedNumber*. Jako nápověda jim slouží upravené kódy pro přijímač a vysílač. Kód pro vysílač nyní odesílá 0, pokud bylo stisknuto tlačítko A a 1, pokud bylo stisknuto tlačítko B. Kód pro přijímač na displeji vypisuje hodnotu proměnné *receivedNumber*. Poté, co žáci přijdou na hodnotu *receivedNumber* je čeká úloha, ve které budou muset kód na přijímači upravit tak, aby se na displeji vypsalo, které tlačítko bylo stisknuto.

Úlohy v další hodině se zabývají kódováním a šifrováním. Upravují poslední zmíněný program tak, aby přes něj šlo posílat zprávu napsanou pomocí

Morseovy abecedy. Také se pokusí jednoduchou zprávu zakódovanou pomocí Morseovy abecedy poslat. Následně kód dále upravují. Například aby bylo poznat, kde v zakódované zprávě končí jedno slovo a začíná druhé, nebo jasně dané ukončení přenosu.

V poslední hodině této kapitoly si žáci s pomocí učitele sestaví systém pro zpracování ankety. Jeden hlavní micro:bit bude sloužit k zahájení ankety a ke sbírání hlasů. Ostatní micro:bity připojené na stejném kanále mohou po obdržení povolení od hlavního micro:bitu hlasovat pro dvě možnosti (pomocí tlačítek A a B). Každý micro:bit může hlasovat jen jednou.

6.3.6 Kapitola 6

Všechny úlohy v této kapitole nepodlehly velkým změnám. Žáci si nejdříve vyzkouší připojení jednoduché trojbarevné LED diody a její rozsvícení a zhasnutí.

Ve druhé hodině pomocí této diody vytvoří magickou lampu. Zde dochází ke změně oproti originálu. Zatímco v původní učebnici se barvy na lampě postupně rozsvěčují a zhasínají, v této verzi se barva vždy plně rozsvítí a před rozsvícením jiné barvy zhasne. Tato změna je zde kvůli dlouhému kódu pro postupné rozsvěcování a zhasínání.

Práce s teplotním čidlem v poslední hodině zůstala nezměněna. Žákům je představen program, který měří teplotu pomocí externího teplotního čidla. V další úloze pak mají žáci za úkol využít zabudované čidlo a zjistit, jak moc se liší hodnoty tohoto a externího čidla.

6.3.7 Závěrečný projekt

Závěrečné projekty jsou převzaty beze změny. V původní učebnici jsou oba závěrečné projekty umístěny uprostřed závěru. V nové verzi učebnice jsem je umístil do nové kapitoly před úplným závěrem.

6.3.8 Žákovské listy

Žákovské listy byly upraveny tak, aby odpovídaly novým metodickým listům. Jinak jejich struktura nebyla změněna.

6.3.9 Samostudium

Samostudium bylo také upraveno podle nových metodických listů. Byla odstraněna nadbytečná teorie. Také byly podrobněji popsány některé programy.

7 Finální verze nové učebnice

7.1 Důvody pro další změnu učebnice

Po obdržení zpětné vazby od autorů původní učebnice došlo k několika zásadním změnám v učebnici. Učebnice by mohla být pořád relativně obtížná pro cílovou skupinu žáků, proto byly do všech kapitol přidány další úlohy. Jejich cílem bylo další procvičení již prezentovaných funkcí nebo omezení skoků v obtížnosti mezi jednotlivými úlohami. Došlo také k přidání seznamovacích úloh na úvody vybraných kapitol. Některé úlohy byly úplně odstraněny a nahrazeny jinými, jiné jen zjednodušeny nebo přepsány, aby byly pro mladší žáky více atraktivní.

Metodika v původní učebnici byla až na teoretické poznatky velmi chudá a spoléhala na učitelovu dobrou znalost prostředí MakeCode. Z toho důvodu byla ve finální verzi více upravena, doplněna a propracována. Například byla doplněna o návodné otázky pro učitele, jejichž zodpovězení může dovést žáky k řešení úlohy. Obtížnější úlohy jsou nyní v metodických poznámkách řešeny krok po kroku.

7.2 Obecné změny v učebnici

V této verzi došlo k odstranění seznamu pomůcek pro každou hodinu. Místo toho byl pro každou kapitolu vypracován jeden seznam pomůcek, který byl umístěn na začátku každé kapitoly.

Struktura učebnice byla změněna. Všechny metodické a žákovské listy byly sloučeny do samostatných souborů. Verze učebnice pro samostudium byla smazána, protože jsem nepředpokládal, že by byla žáky využita.

Vzhledem k velkým úpravám v úlohách došlo k úpravě časové dotace, čímž se uvolnilo místo pro nové úlohy.

Všechny úlohy byly očíslovány, aby byla orientace v učebnici pro učitele jednodušší. Tam, kde to bylo relevantní, jsem přidal zmínku o druhé verzi

micro:bitu, a jak s ním v určitých hodinách nakládat jinak než s první verzí.

7.3 Popis změn po kapitolách

Jako v odpovídající sekci minulé kapitoly, i zde dojde k popisu podrobnějších změn při dalším upravování učebnice.

7.3.1 Kapitola 1

Na začátek první hodiny byl přidán návod na práci s MakeCodem a nahráváním hotových programů do micro:bitu. Na její konec byla přidána úloha, ve které si žáci vyzkouší postupné zobrazení jejich jména po jednom písmenu.


V další hodině bylo přidáno další možné řešení pro úlohu s cyklem s pevným počtem opakování. Toto řešení odpovídá programu, které žáci mohli v minulosti použít v prostředí Scratch. Také bylo přidáno víc úloh týkající se cyklů. Například úloha, ve které mají žáci za úkol napsat takový program, který na micro:bitu vypíše postupně všechna sudá čísla menší než 8 nebo úloha, ve které se na micro:bitu vypíší sestupně všechna lichá čísla od 0 do 10.

Ze třetí hodiny byla odstraněna úloha s úderem srdce. Místo ní bylo zahrnuto několik úloh na procvičování ovládnutí jednotlivých diod pomocí souřadnic. Mezi tyto úlohy patří například úloha, ve které žáci postupně rozsvěcují na micro:bitu určitý tvar nebo písmeno nebo rozsvítí určitý tvar a ten pak postupně zhasnout. Také před tyto úlohy přibyla objevovací úloha, ve které žáci mají sami přijít na funkci bloku *rozsviř*, který ovládá jednotlivé diody.

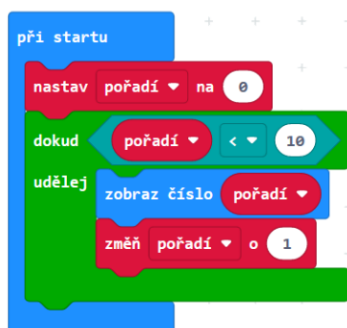
Ve čtvrté hodině byla přidána objevovací úloha na funkci bloku *přepni*. Na konci hodiny byla přidána úloha, ve které mají žáci za úkol postupně rozsvítit a pak zhasnout čtverec. Presentované řešení této úlohy bylo ponecháno neoptimalizované, proto mají rychlejší žáci prostor pro vlastní úpravy. Zbytek úloh zůstal, ale k některým byly přidány další možná řešení, které by žáci vymyslet.

7.3.2 Kapitola 2

V jediné hodině této kapitoly došlo jen k malým úpravám. Bylo přidáno další možné řešení pro úlohu s počtem zmáčknutí tlačítka A a B. K poslední úloze, která byla v původní učebnici neřešená, bylo poskytnuto ukázkové řešení.

 **5. úkol.** Žáci mají vytvořit stejně se chovající program, který vypisuje čísla od 0 do 9, ovšem pomocí bloku dokud. Necháme tak žáky procvičit si blok s podmínkou pro jeho ukončení (cyklus while).

V Makecode je logika bloku dokud jiná než ve Scratchi. V Makecode se bloky uvnitř vykonávají, dokud podmínka platí, tedy dokud nepřestane být splněna. Ve Scratchi je tomu naopak – tam se opakuje až do doby, kdy podmínka nezačne být splněna.



Obrázek 12: Ukázka úlohy ve finální verzi učebnice

7.3.3 Kapitola 3

V této kapitole kromě menších úprav v zadání nedošlo k žádným změnám

7.3.4 Kapitola 4

Na začátku kapitoly byly obrázky z původní učebnice, které ukazují směry tří os, které micro:bit sleduje, nahrazeny výčtem všech možných gest, které micro:bit zvládá. Z toho důvodu také byla buďto přepsána nebo úplně smazána teorie týkající se tří os. Úlohy, ve kterých se na micro:bitu při nějaké události objevilo písmeno, byly upraveny tak, aby micro:bit vykreslil místo písmen obrázky. První úloha pak byla rozdělena na tři různé úlohy. V první žáci objeví, jak gesta pro naklonění micro:bitu fungují, ve druhé vytvoří prototyp

vodováhy, a ve třetí úloze tento prototyp otestují a upraví tak, aby ukazoval, kdy je micro:bit nakloněn vodorovně. Na konec hodiny byla přidána úloha, ve které žáci mají popustit uzdu svojí kreativitě a vymyslet využití jednoho gesta v reálném životě.

Do druhé hodiny byla přidána objevovací úloha, ve které mají žáci vyzkoušet funkčnost generátoru náhodných čísel. Na tuto úlohu navazuje úloha vytvoření hry Kámen, nůžky, papír z předchozí verze učebnice. V této verzi ale došlo k podrobnějšímu popisu programu. Na konci hodiny došlo k odstranění společného řazení. Tato aktivita byla nahrazena vytvořením programu pro náhodné generování telefonních čísel, hesel a podobně.

Ve třetí hodině došlo k upravení textů zadání, aby více odpovídaly současné vizi učebnice. Do úlohy, ve které mají žáci za úkol sestavit program rozlišující východ a západ byl přidán obrázek kompasu s vyznačeným intervalem, ve kterém bude micro:bit nakloněn na východ.

Autoři původní učebnice se báli, jestli je teorie a program zabývající se intenzitou magnetického pole vhodná pro žáky základních škol. Proto byla tato úloha nahrazena mnohem lehčí variantou na intenzitu světla. To je další veličina, kterou micro:bit může měřit. Z toho důvodu byly všechny texty úplně přepracované. Také byla hodina rozdělena do dvou úloh. V první úloze děti mají sami přijít na funkci bloku *intenzita světla*. Ve druhé úloze pak žáci naprogramují micro:bit tak, aby určil, zda je den nebo noc.

7.3.5 Kapitola 5

Na začátek této kapitoly byl přidán návod jak nastavit rádio pro každou skupinu žáků. Do první hodiny byla také zahrnuta upravená verze úloha s počítadlem na dálku.

Ve druhé kapitole došlo k malým úpravám v textu. Byla přidána úloha na začátek hodiny, aby si žáci více procvičili posílání zpráv mezi micro:bity. Úloha na rozlišení stisknutého tlačítka byla doplněna o další možné řešení a náměty pro rychlejší žáky.

Ve třetí hodině byly více rozvedeny úlohy na hledání problémů s vysíláním Morseovy abecedy. Byly k nim vytvořeny nové texty a obrázky a upravena časová dotace tak, aby žáci na provedení jednotlivých kroků měli dostatek času. Na konec hodiny byl doplněn kompletní kód tvořený během hodiny.

Poslední hodina této kapitoly byla celá přepracována. Zabývá se stejnou úlohou vytvoření anketního systému, ta je ale rozdělena do několika dílčích kroků, které jsou podrobně popsány. Také na tuto aktivitu byla přidána jedna celá hodina do celkové časové dotace kapitoly. Všechny kroky řešení jsou doplněny o otázky, které by měly nasměrovat žáky ke správnému řešení.

7.3.6 Kapitola 6

Začátek kapitoly byl doplněn o rozsáhlou teorii, která je určena výhradně pro učitele a seznamuje je s připojováním periférií k micro:bitu. Také vysvětluje jaké hodnoty může micro:bit číst ze svých pinů a jak s nimi nakládat. Tato teorie není určena pro žáky.

Na začátku první hodiny je přidána úloha, ve které si žáci vyzkouší základní rozsvícení a zhasnutí jedné barvy diody pomocí tlačítek. Také byla přidána úloha, ve které dioda několikrát sama zabliká jednou barvou.

Úloha s magickou lampou byla také kvůli svojí komplexnosti podrobně popsána a každá část kódu pořádně vysvětlena. Bylo přidáno pár tipů, jak tuto úlohu s žáky řešit.

Na konec kapitoly byla přidána další hodina. Do ní je přesunut jeden ze závěrečných projektů a je krok po kroku s žáky vyřešen. Tento projekt se tématicky hodil více do této kapitoly, a navíc to dovoluje mít jen jeden závěrečný projekt místo dvou.

7.3.7 Závěrečný projekt

Závěrečný projekt je nyní jen jeden. i přes to se jedná o volný projekt, který mohou žáci pojmout po svém. Do této kapitoly bylo přidáno ukázkové obecné

řešení, které slouží jako šablona pro učitele a ukazuje, jak by možné řešení mohlo vypadat. Funkčnost tohoto řešení byla podrobně popsána.

7.3.8 Žákovské listy

Žákovské listy prošly velkou úpravou. Bylo z nich smazáno vše, až na zadání úloh, protože během informatiky je důležité rozvíjení žáků pomocí samostatné práce místo prezentování teorie. Smazána tedy byla veškerá teorie, návody a postupy k řešení a i samotná správná řešení, která v původní učebnici sloužila k opisování kódu místo jeho vymyšlení. Díky této změně nebudou žáci zatěžováni teorií, neuvidí řešení jednotlivých úloh a mohou pracovat vlastním tempem. Všechny žákovské listy byly spojeny do jednoho velkého souboru, který může učitel žákům vytisknout.

3. úkol

Vyzkoušejte, co dělá blok při zatřesení. Sestavte další scénář, který zobrazí jinou ikonu.

Kdy se tato událost dá využít?

4. úkol

Vytvořte program, který bude neustále na displeji zobrazovat, kolikrát bylo stisknuto tlačítko A.

Před vytvářením programu odpovězte na následující otázky:

1. Jak si bude micro:bit pamatovat, kolikrát se tlačítko stisklo?
2. Jak zařídit, aby na displeji svítila zapamatovaná hodnota?
3. Jak naprogramovat, co se stane po stisknutí tlačítka?

Na konec přidejte scénář, který po stisknutí tlačítka B naopak jedničku odečte.

K čemu by se takto naprogramovaný micro:bit dal použít?

Obrázek 13: Ukázka zadání úloh v žákovských listech

7.4 Vydání učebnice

S těmito úpravami byla učebnice zkompletována a vydána pod názvem Robotika pro základní školy: programujeme micro:bit pomocí Makecode.

Učebnice se skládá ze dvou souborů formátu pdf. První soubor v sobě obsahuje všechny metodické poznámky určené pro učitele rozdělené podle kapitol. Druhý soubor je určen pro žáky a obsahuje zadání všech úloh a závěrečného projektu rozdělených podle kapitol.

Učebnice je volně dostupná ke stažení na portálu iMyšlení. Odkaz ke stažení: <https://tinyurl.com/UcebniceMakeCode>

Kompletní učebnice je také dostupná jako příloha elektronické verze této práce a na CD přiloženém u fyzické verze této učebnice.

8 Závěr

Byla vytvořena učebnice robotiky pomocí programovací desky micro:bit pro druhý stupeň základní školy. Učebnice vznikla postupnými úpravami a předělvkami učebnice *Robotika pro střední školy: programujeme Micro:bit pomocí Pythonu*, která je primárně určena pro technicky zaměřené střední školy. Nejdříve došlo k prostému předělání úloh a textů z prostředí Python do blokového prostředí MakeCode a k doplnění nových úloh tak, aby se snížila obtížnost mezi navazujícími úlohami.

Úlohy z původní učebnice byly otestovány a nevyhovující úlohy byly nahrazeny jinými úlohami. Také došlo k doplnění nových úloh na místa, kde v původní docházelo k velkým skokům v obtížnosti mezi jednotlivými úlohami. Všechny úlohy byly upraveny tak, aby žákům jen nepředkládaly hotová řešení. Jsou k nim například přiloženy návodné otázky pro učitele, pomocí kterých může učitel navést žáky ke správnému řešení.

Některé z úloh, které by mohly být pro žáky atraktivní, jako například úloha s magickou lampou, se bohužel nepovedlo úplně do blokového prostředí převést, proto musely být upraveny a jejich současné verze nemusí být pro žáky tolik atraktivní. V kapitole Síť byly také ze stejného důvodu odstraněny úlohy, ve kterých se propojují dva micro:bity pomocí kabelů. Žákům je tedy jen ponechána možnost je propojit bezdrátově.

Dále došlo k popsání a porovnání vybraných dostupných blokových prostředí a byla posouzena jejich vhodnost při programování micro:bitu a využití ve výuce. Na základě tohoto porovnání bylo vybráno prostředí Microsoft MakeCode, ve kterém došlo k realizaci úloh v učebnici.

Proběhla revize české mutace Microsoft MakeCode. Český překlad byl upraven tak, aby odpovídal blokovému prostředí Scratch. Kvůli nemožnosti překladu některých textů bohužel není tento překlad kompletní. Nepřeložené části ale nevadí při využití prostředí ve výuce. Jedná se o základní anglické výrazy, které už žáci s největší pravděpodobností znají. V učebnici se pracuje jen s jed-

ním anglicky nadepsaným blokem, a to s blokem receivedNumber v kapitole Síť.

O vytvoření této učebnice a dalších učebnic z projektu PRIM bylo referováno 1.4. 2021 na konferenci Počítač ve škole.

Učebnice byla vydána pod licencí Creative Commons a je volně dostupná na internetovém portálu iMyšlení. Odkaz ke stažení učebnice je <https://tinyurl.com/UcebniceMakeCode>. Kompletní text učebnice včetně metodických listů a žákovských listů je dostupný jako příloha elektronické verze této práce a na přiloženém CD u fyzické verze učebnice.

Seznam použité literatury a zdrojů

- [1] DENNING, Peter J. a Matti TEDRE. *Computational Thinking*. Cambridge: The MIT Press, 2019. ISBN 978026253656.
- [2] LESSNER, Daniel. Analýza významu pojmu „Computational Thinking“ In: *Journal of Technology and Information Education* [online]. 2014, 6 (1), pp. 71-88 [vid. 15.2.2021]. ISSN 1803-537X. Dostupné z: https://jtie.upol.cz/artkey/jti-201401-0006_ANALYZA_VYZNAM_POJMU_COMPUTATIONAL_THINKING.php
- [3] *Co je informatické myšlení* [online]. Jihočeská univerzita v Českých Budějovicích. [vid. 15.2.2021]. Dostupné z: <https://imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>
- [4] SELBY, Cynthia C. a John WOOLLARD. *Refining an Understanding of Computational Thinking*. Southampton: University of Southampton, 2014. [vid. 8.3.2021]. Dostupné z: <https://eprints.soton.ac.uk/372410/>
- [5] PECH, Jiří a Milan NOVÁK. *Robotika pro střední školy: programujeme Micro:bit pomocí Pythonu* [online]. České Budějovice: Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta. [vid. 10.2.2021]. ISBN 978-80-7394-787-3. Dostupné z: <https://imysleni.cz/ucebnice/robotika-ucebnice-pro-stredni-skoly-micro-bit>
- [6] SENTANCE, Sue, Jane WAITE, Steve HODGES, Emily MACLEOD a Lucy YEOMANS. "Creating Cool Stuff"- Pupils'experience of the BBC micro:bit. In: *Proceedings of the 48th ACM Technical Symposium on Computer Science Education: SIGCSE 2017* [online]. New York: Association for Computing Machinery, 2017. [vid. 10.2.2021]. Dostupné z: <https://dl.acm.org/doi/10.1145/3017680.3017749>

- [7] *Meet the new BBC micro:bit* [online]. Micro:bit Educational Foundation [vid. 10.2.2021]. Dostupné z: <https://microbit.org/new-microbit/>
- [8] CELLAN-JONES, Rory. The Micro Bit - can it make us digital? In: *BBC News* [online]. British Broadcasting Corporation, 2015. [vid. 10.2.2021]. Dostupné z: <https://www.bbc.com/news/technology-31859283>
- [9] British Broadcasting Corporation. About Make It Digital. In: *BBC Make It Digital* [online]. British Broadcasting Corporation, 2015. [vid. 10.2.2021]. Dostupné z: <https://www.bbc.co.uk/programmes/articles/1gkww58DPmRzt2TzDp3pr9x/about-make-it-digital>
- [10] ROCKS, Sinead. BBC micro:bit at Bett 2016 In: *BBC Blog* [online]. British Broadcasting Corporation, 2016. [vid. 10.2.2021]. Dostupné z: <https://www.bbc.co.uk/blogs/aboutthebbc/entries/32f2edad-d51f-41bb-af98-f42f41d0a705>
- [11] Top Five Coding Languages for Kids in 2019. In: *MakeBlock* [online]. MakeBlock, 2019. [vid. 12.2.2021]. Dostupné z: <https://www.makeblock.com/official-blog/245845.html>
- [12] Using the micro:bit with Scratch. In: *micro:bit Help & Support* [online]. Micro:bit Educational Foundation, 2020. [vid 12.2.2021]. Dostupné z: <https://support.microbit.org/support/solutions/articles/19000026080-using-the-micro-bit-with-scratch>
- [13] *Let's Code* [online]. Micro:bit Educational Foundation [vid. 12.2.2021]. Dostupné z: <https://microbit.org/code/>

Seznam obrázků

1	Žákovské zadání ukázkových úloh	16
2	Programovatelná deska BBC micro:bit	18
3	Prostředí Scratch	19
4	Prostředí Microsoft MakeCode	20
5	Prostředí MicroBlocks	21
6	Prostředí Vittascience	21
7	Prostředí Mu Python Editor	24
8	Ukázka prostředí Crowdin	26
9	Ukázka nevyhovující úlohy v původní učebnici	28
10	Ukázka přidané úlohy	36
11	Ukázka upravené úlohy	38
12	Ukázka úlohy ve finální verzi učebnice	44
13	Ukázka zadání úloh v žákovských listech	47

A Příloha 1: Metodické listy učebnice pro učitele

Uvedené metodické listy jsou také dostupné na příloženém CD nebo online na portálu iMyšlení na adrese
<https://tinyurl.com/UcebniceMakeCode>

ÚVOD

Tato učebnice si klade za úkol dát učitelům a žákům do rukou materiál, s jehož pomocí se naučí základy a principy robotiky a elektrotechniky pomocí jednočipové vývojové platformy **BBC micro:bit**¹. Současně aplikuje základní znalosti programování a algoritmizace. Ke zvládnutí učebnice **jsou třeba znalosti základů programování** v blokovém prostředí, např. Scratch.

Tato učebnice vznikla přepracováním původní učebnice **Robotika pro střední školy: programujeme Micro:bit pomocí Pythonu** autorů Jiřího Pecha a Milana Nováka. Využívá blokové programovací prostředí **Microsoft MakeCode** (<https://makecode.microbit.org>) a je poskytována pod licencí CC-BY-SA (uveďte původ – zachovejte licenci). Učebnice je upravena pro úroveň žáků 8. – 9. ročníků základní školy nebo žáky středních škol netechnického zaměření.

Učebnice je stavěna tak, aby žáci v naprosté většině lekcí a příkladů vystačili pouze s micro:bitem a nemuseli sestavovat žádné obvody. Výjimkou je připojení reproduktoru (sluchátek) pomocí kabelů s krokodýlky pro přidání audio výstupu ve třetí kapitole a volitelné aktivity pro připojení externích součástek v poslední kapitole.

Micro:bit má výhodu, že jej lze programovat v různých jazycích a mezi nimi přitom volně přecházet. K této učebnici existuje výše citovaná varianta pro programování micro:bitu v jazyce Python, která má téměř identický obsah. Je tedy možné, že pokročilejší žáci se zájmem o programování v „dospělejší“ jazyce by mohli po několika kapitolách přejít na odpovídající kapitolu učebnice programování v jazyce Python.

STRUKTURA UČEBNICE

Učebnice má následující části:

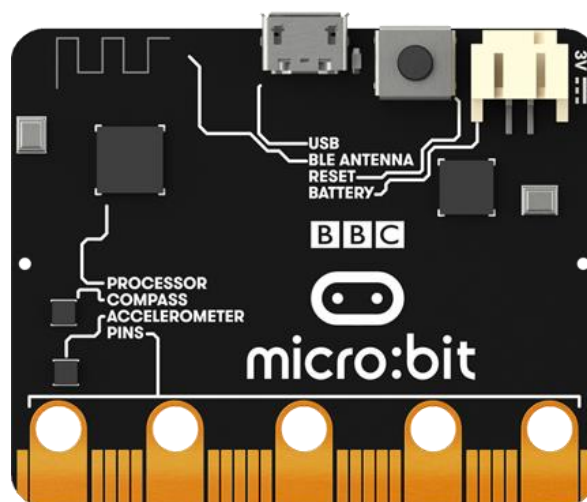
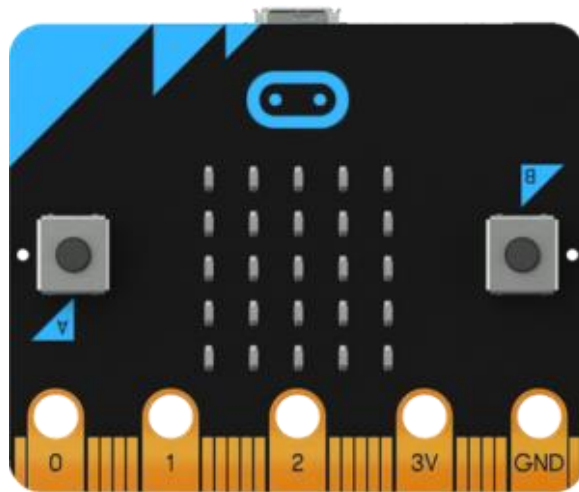
1. **Pro učitele** – obsahuje průvodce vyučovacími hodinami s metodickými poznámkami, jak vést výuku a na co klást důraz. Obsahuje vzorová řešení úloh, seznam potřebného materiálu a odhad potřebného času pro výuku.
2. **Pro žáky** – soubor pracovních listů se zadáními všech úloh, rozdělenými podle kapitol a očíslovanými stejně jako v učitelových průvodcích hodinou. Pracovní listy je možné rozdat žákům, aby měli zadání úloh k dispozici a mohli při učení udržovat individuální tempo.

Tato učebnice není členěna do kapitol podle programovacích konceptů nebo složitosti programování, ale podle součástí desky micro:bit, které budou žáci zkoumat a ovládat pomocí programu, který sestaví. Začíná se výbavou samotné desky, později se k ní připojují další součástky.

1 Micro:bit je ochranná známka Micro:bit Educational Foundation

CO JE MICRO:BIT

Micro:bit je open-source vývojový kit vyvinutý ve Velké Británii za podpory BBC určený primárně pro výuku informačních technologií.



Micro:bit obsahuje:

- 5x5 matici LED diod
- dvě programovatelná tlačítka (označení A a B)
- kompas
- tříosý akcelerometr (gyroskop)
- 17 GPIO pinů, z nich 3 snadno přístupné pomocí např. krokodýlových kabelů

Dále umožňuje:

- zjišťovat intenzitu osvětlení, magnetického pole a teplotu
- komunikaci pomocí Bluetooth

- přímou komunikaci pomocí rádia
- připojení sluchátek či reproduktoru

Micro:bit je možné programovat pomocí několika programovacích jazyků - **Microsoft MakeCode** (grafický jazyk podobný Scratchi) **JavaScript** a **MicroPython**. Mezi MakeCode a JavaScriptem nebo Pythonem lze při programování přepínat a kombinovat je, na druhou stranu je nutné být online.

CO BUDETE POTŘEBOVAT

- **BBC micro:bit** pro každého žáka nebo pro dvojici žáků.
- USB kabel s micro USB zakončením. Pokud nebude váš USB kabel fungovat, vyzkoušejte jiný, stává se to (existují kabely pouze pro napájení, nikoliv pro přenos dat). Obecně platí, čím kratší kabel, tím lépe bude fungovat pro datový přenos.
- Micro:bit můžete napájet buď prostřednictvím zmíněného USB kabelu, nebo potřebujete držák na baterie (obvykle dvě AAA) s odpovídajícím kabelem (dodává se jako součást sady Micro:bit Go)
- Počítač s libovolným operačním systémem (Windows, Linux, Mac OS, Chrome OS) a internetovým připojením. S **Makecode** budeme vždy pracovat na stránkách <https://makecode.microbit.org/>.
Existuje možnost programovat Micro:bit pomocí Scratch, tato možnost ovšem postrádá velmi dobrý softwarový simulátor micro:bitu.

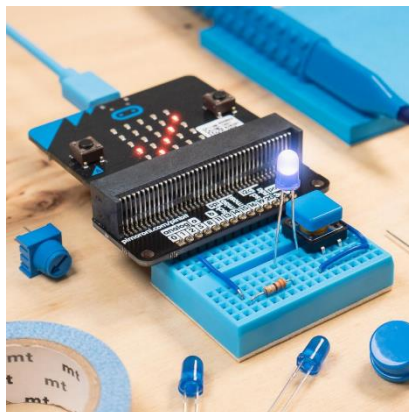
Pouze výjimečně bude potřeba další vybavení:

- Pokud chcete, aby v kapitole 3 váš micro:bit vydával zvuky, budete potřebovat libovolný reproduktor (sluchátka) vybavený běžným konektorem jack.
- Dva vodiče. Ideální jsou vodiče osazené na obou koncích „krokodýly“. Viz obrázek:



- Pokud chcete připojovat různá externí zařízení (LED, teploměr atd.) poříd'te si těchto vodičů více – cca. čtyři na jeden micro:bit v různých barvách. Z toho jeden by měl být červený (pro plus) a jeden černý (pro minus – zem).

- Vodiče s krokodýlky můžete nahradit nepájivým polem a propojovacími vodiči.



- Pokud budete probírat i kapitolu 6 – Periférie, potřebujete následující součástky:
 - Tříbarevnou diodu se společnou katodou (zem).
 - Teplotní čidlo pracující s napětím 3 V, např. TMP36.

PŘEDPOKLÁDANÉ VSTUPNÍ ZNALOSTI

K zvládnutí učebnice jsou třeba znalosti základů programování v blokovém prostředí, např. Scratch.

Co se týče znalostí elektroniky a zapojování obvodů, nejsou žádné speciální znalosti vyžadovány, vše je probíráno od základů.

ČASOVÁ DOTACE A TÉMATA

Celá učebnice je navržena na celkem 20 – 23 vyučovacíh hodin. Minimální časová dotace při vynechání poslední rozšiřující kapitoly a dobrovolných úkolů je 16 hodin. Při realizaci rozšiřujících projektů může spotřebovat i více než uvedených 23 hodin.

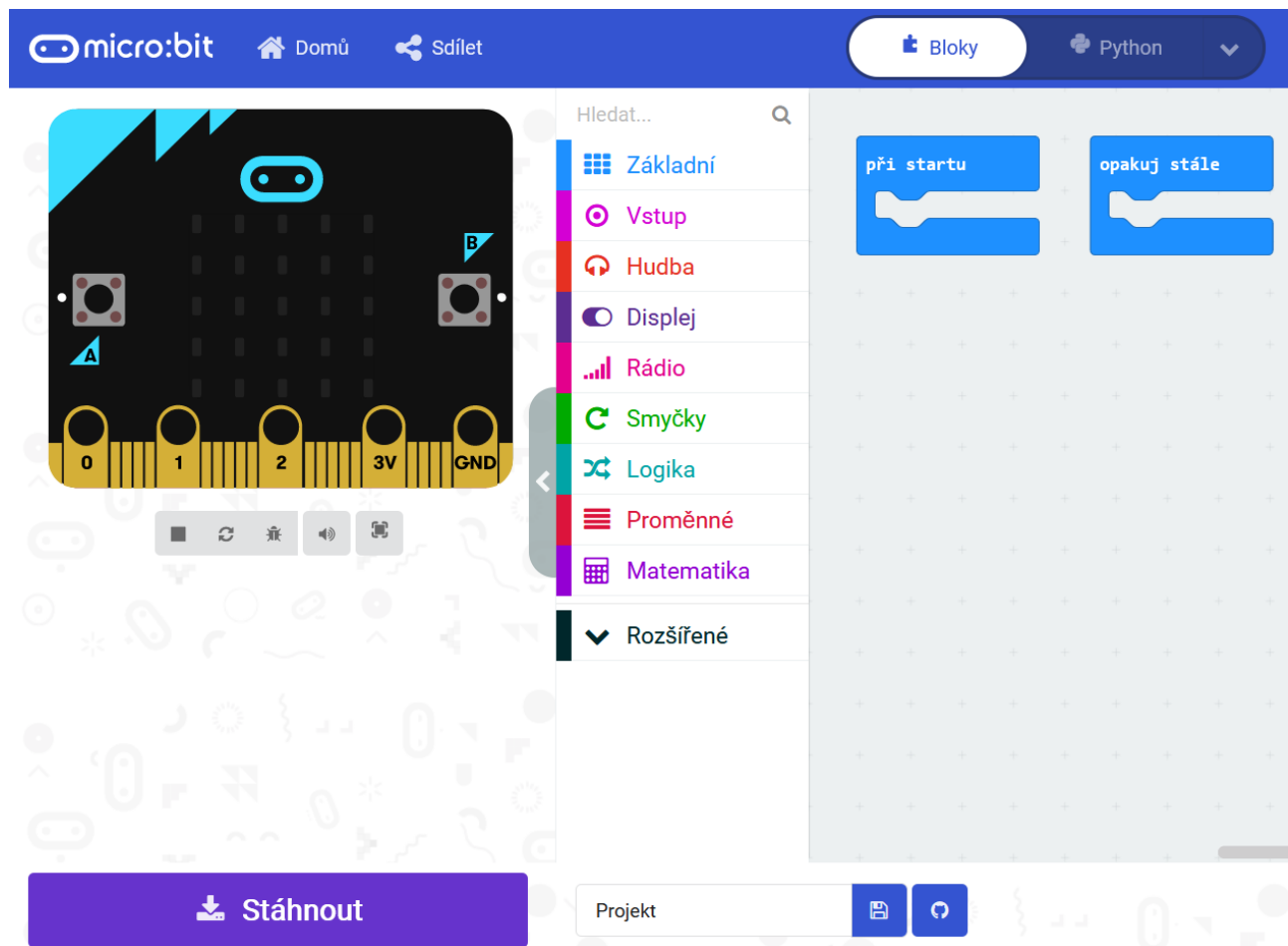
Učebnice se skládá ze 6 témat rozdělených do kapitol. Každá kapitola je členěna na vyučovací hodiny, každá vyučovací hodina na jednotlivé kroky s uvedenou spotřebou času.

Přehled kapitol:

1. VÝPIS NA LED DISPLEJ
2. PRÁCE S TLAČÍTKY
3. HUDBA
4. POLOHA
5. SÍŤ
6. PŘIPOJOVÁNÍ PERIFÉRIÍ

PROGRAMOVACÍ PROSTŘEDÍ MAKECODE

Přidáváme ukázkou prostředí **MakeCode** spolu s vysvětlením jeho struktury:



Na levé straně vidíte simulaci micro:bita, která slouží k testování vašich programů. Díky ní není nutno každý program nahrávat do micro:bita; tato simulace totiž dokáže kód sama vykonat.

Na pravé straně vidíte plochu pro svoje scénáře, pro svůj program. Pro přidání bloku do scénáře (nebo přidání dalšího scénáře) stačí přetáhnout blok z nabídky (uprostřed obrazovky) na pravou stranu okna aplikace. Některé bloky jsou v rozšířené nabídce dole.

S bloky se dá manipulovat podobně jako v programovacím prostředí **Scratch**. Skládají se pod sebe nebo se do sebe vnořují. Na rozdíl od Scratche nelze kliknutím spustit izolovaně jeden scénář.

Použitá terminologie:

Blok – grafický objekt „obdélník“, obsahuje jeden příkaz, jeden „řádek“ programu

Scénář – více bloků pospojovaných do jednoho celku

Program – jeden nebo více scénářů; vše, co je na ploše a co se „posílá“ do micro:bitu

1. VÝPIS NA LED DISPLEJ

Co se naučíte

- Ovládat matici 5x5 LED diod na micro:bitu
- Zobrazit běžící text nebo jeden statický znak
- Zobrazit přednastavený obrázek
- Vytvořit jednoduchou animaci

Co budete potřebovat

- PC s internetovým připojením
- Propojovací USB kabel micro USB koncovkou
- Micro:bit

Časová náročnost

4 až 5 vyučovacích hodin po 45 minutách

PRŮVODCE HODINOU I-1

Žáci se seznámí s *micro:bitem* a programovacím prostředím *MakeCode*.

Co bude v této hodině potřeba:

- PC s internetovým připojením. Na něm si žáci otevřou programovací prostředí *MakeCode*. Stránky editoru jsou <https://makecode.microbit.org/>
- Micro:bit s USB kabelem zakončeným micro USB. Pozor nefungují všechny kabely. Pokud budete používat jiné než koupené spolu s *micro:bitem*, je nutné je předem vyzkoušet.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro žáky

1. krok 10 minut

Vysvětlíte žákům, co se naučí v tomto předmětu:

- Poznájí výukovou platformu *micro:bit* a naučí se jí ovládat
- Naučí se některým principům z elektronických obvodů
- Naučí se základy programování

Rozdejte žákům pracovní listy a *micro:bity*.

Krátce žákům *micro:bit* popište (displej, tlačítka, napájení, konektor USB). Nechte žáky, aby si jej prohlédli. Následující vlastnosti jsou zde pro úplnost učebnice vypsány, ale není úplně nutné, aby je žáci znali:

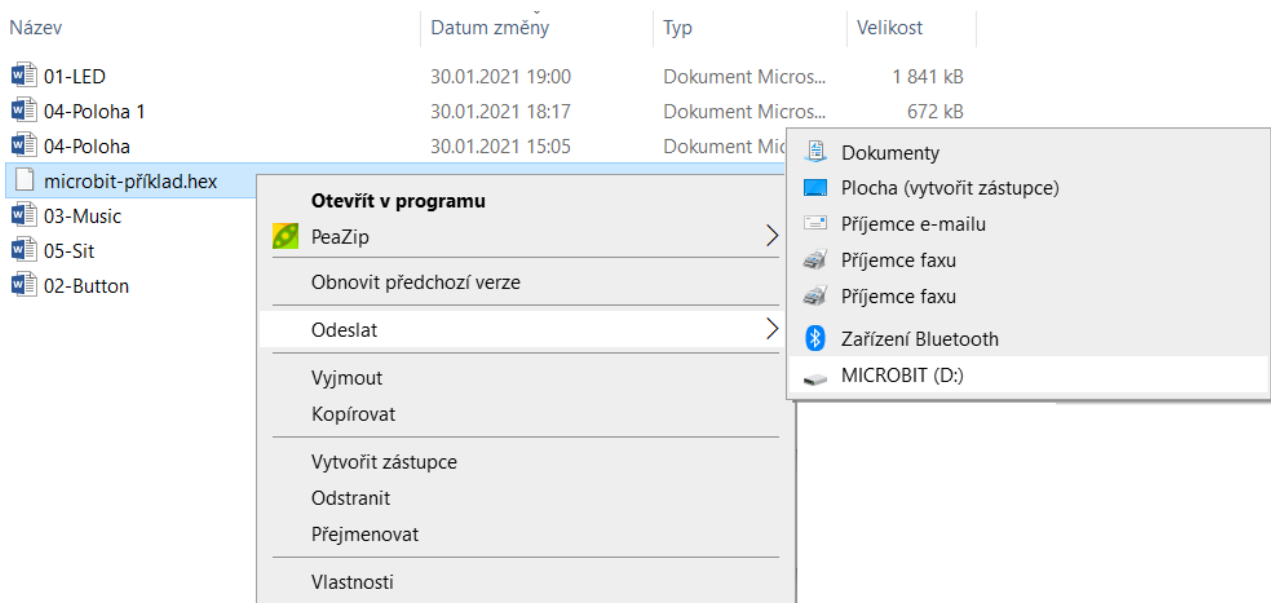
- displej 5x5 diod
- dvě programovatelná tlačítka zepředu, tlačítko reset vzadu
- vstupy – micro USB a napájecí konektor
- akcelerátor, magnetometr
- možnost měřit intenzitu osvětlení a teplotu (ale obojí ne zcela přesně)
- připojení k mobilu přes bluetooth
- možnost vzájemné komunikace mezi *micro:bity*
- na dolním okraji 17 pinů pro přenos dat při připojení vnějších zařízení, např. senzorů nebo servomotorů (GPIO - general-purpose input/output), dále piny 3V a GND. Tři piny jsou větší (označené 0, 1, 2) pro snadné připojení vodičů s krokodýlky. Ostatní piny pro plnohodnotné použití vyžadují speciální základnu.

2. krok 10 minut

Představte editor programu pro micro:bit *MakeCode* a jeho ovládání, použijte projekci. Pokud žáci nepracovali ve Scratch, na jednoduchém příkladu krátce vysvětlíte princip přetahování a spojování bloků. Vhodným příkladem je program na zobrazení textu „Ahoj“ ze 4. kroku.

Ukažte, jak se program zkopíruje do micro:bitu. Ukažte také, že v Makecode funguje tzv. simulátor skutečného micro:bitu.

Po uložení projektu do souboru (tlačítkem Stáhnout) najdete soubor v souborovém manažeru a zkopírujte jej do micro:bitu (který je připojen jako externí disk), nebo soubor do micro:bitu odešlete pomocí pravého tlačítka (viz obrázek).



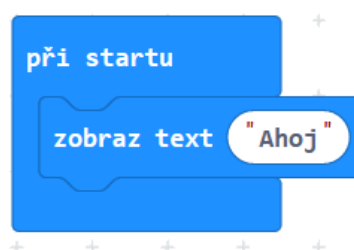
Dále vysvětlíte základní terminologii, která se v učebnici používá. Učebnice využívá stejného označení jako programovací prostředí Scratch. Pod pojmem blok tedy rozumíme jen jeden blok, který vybereme z nabídky bloků. Scénářem pak myslíme několik bloků poskládaných do celku. Program a kód pak označují celý jeden projekt, který obsahuje jeden nebo více scénářů.

3. krok 15 minut



Zadejte žákům **1. úkol** z pracovního listu.

Žáci sestaví svůj první program. V prohlížeči si spustí webové stránky *MakeCode*, (<https://makecode.microbit.org/>), vytvoří si nový projekt a připojí micro:bit USB kabelem. V editoru poskládají bloky do následujícího scénáře:



Nabídka bloků se dělí na sekce. V této hodině budou používat bloky ze sekce **Základní**.

1. VÝPIS NA LED DISPLEJ

Vysvětlete žákům, že v testech **nelze používat znaky s českou diakritikou**, protože je displej micro:bitu neumí zobrazit.

Nechte žáky nahrát program na micro:bit a řešte s nimi případné chyby.

- Program nelze nahrát – zkontrolujte, zda je micro:bit připojený, zkuste jiný kabel, USB port, micro:bit, počítač.
- Micro:bit nic nepíše – zkontrolujte, jestli jste správně poskládali bloky. Bloky, které jsou nefunkční, jsou šrafované a nevybarvené.

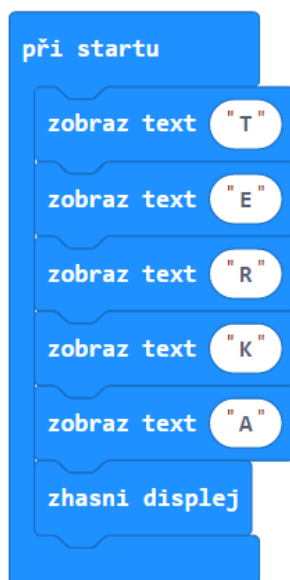
4. krok 10 minut



2. úkol. Žáci dostanou za úkol pospojovat několik bloků. Jejich úkolem je zobrazit jejich jméno tak, aby před očima nepřejelo, ale aby se jednotlivá písmenka střídala.

Jako radu lze zadat úkol pro zobrazení jednoho písmena.

Ukázka řešení úlohy:



PRŮVODCE HODINOU I-2

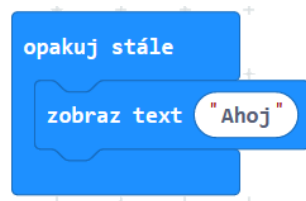
V této hodině budou žáci pracovat s bloky pro opakování – nekonečné opakování, opakování s proměnnou, která počítá počet proběhlých opakování, a opakování s podmínkou jeho ukončení.

1. krok 10 minut



3. úkol. Žáci prozkoumají blok `opakuji stále`. Nechejte je stále zobrazovat nějaké slovo.

Poté se zeptejte na rozdíl mezi tímto blokem a blokem `při startu` z předchozí hodiny.



Žáci mohou vyzkoušet, aby se na displeji opakovalo jejich jméno tak, že se jednotlivá písmenka budou střídat, přepisovat.

2. krok – 15 minut



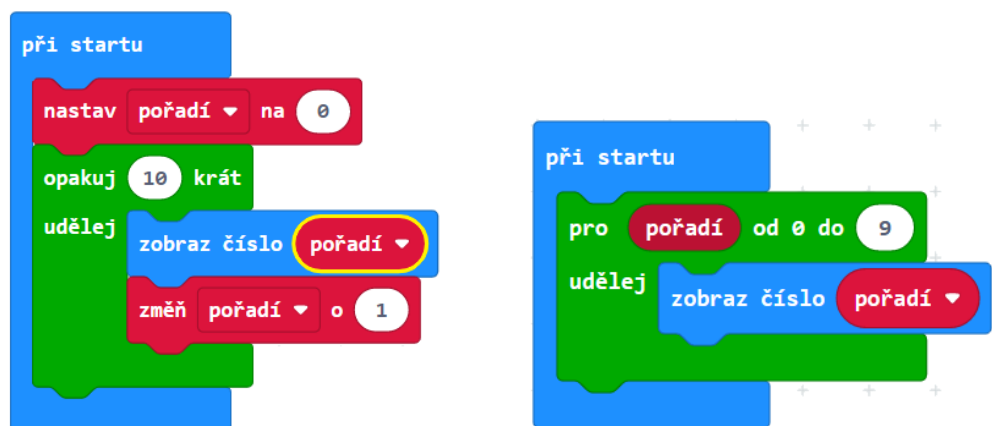
Zadejte žákům **4. úkol** z pracovního listu. Úkolem bude sestavit program, který vypíše čísla od 0 do 9 (volíme tak proto, že větší čísla displejem „protékají“).

Pro zapamatování si, které číslo se má právě vypisovat, budeme potřebovat proměnnou. Toto je možné prozradit žákům jako nápovědu.

Žáci mohou použít blok `opakuji 10 krát`, který znají ze Scratche, a nastavovat a měnit hodnotu proměnné příkazy `nastav` a `změň`. Mohou také použít blok opakování `pro` (cyklus for), do něhož je proměnná již zakomponována.

Žákům pomůže, když jim sdělíte, v jaké sekci nabídky příslušné bloky najdou.

Správné řešení, dvě varianty - použití obou typů bloků pro pevný počet opakování:



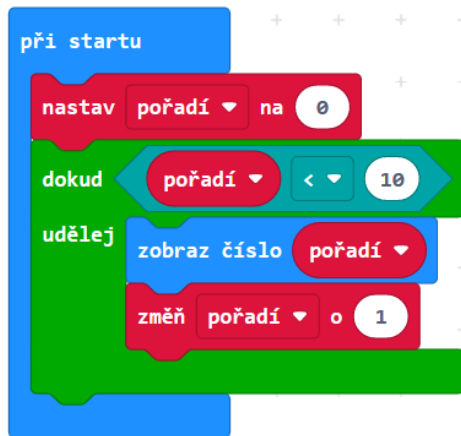
U programu vpravo je dobré vědět, že proměnnou **pořadí** nelze použít mimo blok `pro`.

3. krok – 10 minut

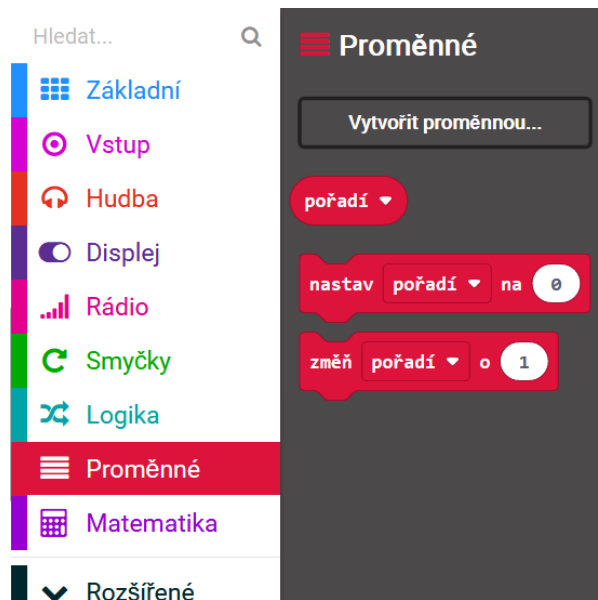


5. úkol. Žáci mají vytvořit stejně se chovající program, který vypisuje čísla od 0 do 9, ovšem pomocí bloku `dokud`. Necháme tak žáky procvičit si blok s podmínkou pro jeho ukončení (cyklus `while`).

V Makecode je logika bloku `dokud` jiná než ve Scratchi. V Makecode se bloky uvnitř vykonávají, dokud podmínka platí, tedy dokud nepřestane být splněna. Ve Scratchi je tomu naopak – tam se opakuje až do doby, kdy podmínka nezačne být splněna.



Bloky při práci s proměnnými se objeví až poté, co proměnnou vytvoříme. Sekce bloků **Proměnné** je po startu aktivity prázdná, ale po vytvoření proměnné vypadá takto:



V samotném scénáři se pak hodnota předem vytvořené proměnné `pořadí` nastaví na 0 a jako podmínka se nastaví `pořadí < 10`. Blok pro tuto i další podmínky se dají najít v sekci **Logika**.

Pozor, kam žáci umisťují bloky `nastav` a `změň`. První z nich musí ležet mimo blok `dokud` a druhý uvnitř něho.

4. krok – 10 minut

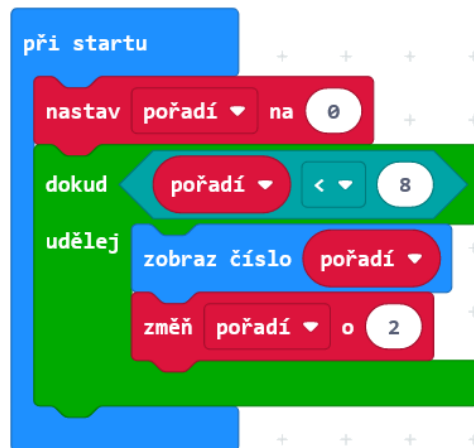


Nechejte žáky řešit **6. úkol** z pracovního listu:

Pomocí cyklů se na displej micro:bitu mají vypsát sudá čísla menší než 8.

Úloha nespecifikuje, který cyklus mají žáci použít, ale nabízí se využít cyklus s neurčitým počtem opakování. Nejslabší žáci možná sestaví scénář bez cyklů.

Řešení:



Zeptejte se žáků, jestli se má při použití tohoto scénáře vypsát i číslice 8? Pokud ne, nechte je, aby scénář upravili tak, aby se číslice 8 vypsala.

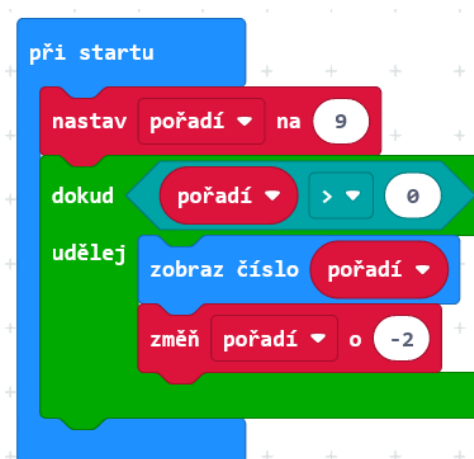
V podmínce je znaménko $<$, které značí „menší než“. Jeho nahrazením za \leq , které značí „menší nebo rovno než“ docílíme toho, že se vypíše i číslice 8.



7. úkol. Nyní zadejte žákům podobnou úlohu:

Vypište všechna lichá čísla větší než 0 a menší než 10 od největšího, po nejmenší.

Řešení:



Vysvětlíte žákům, že hodnota -2 v posledním bloku bude snižovat hodnotu proměnné pořadí o 2.

PRŮVODCE HODINOU I-3

Žáci se seznámí s grafikou na displeji micro:bitu. Vyzkouší si jak zobrazení připravených obrázků, tak tvorbu obrázků vlastních.

1. krok 10 minut

Žáci vyzkoušejí bloky ukaž ikonu a čekej.



8. úkol. Úkolem je, aby se několik obrázků na displeji vystříдалo, a nakonec displej zhasl. Pro delší dobu zobrazení jednoho obrázku použijí čekání.

Je třeba upozornit, že čas je napsán v milisekundách, a zeptat se žáků, jaké číslo má být napsáno, aby program čekal 1 sekundu.

Můžete chtít, aby obrázky dohromady dávaly nějaký smysl, např. micro:bit v této ukázce řešení úlohy střídá nálady:



2. krok 10 minut

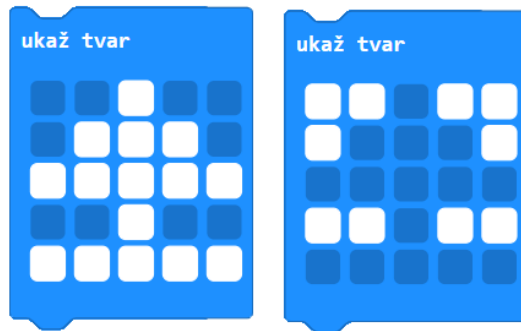
Žáci si vyzkouší vytvoření vlastního obrázku v bloku ukaž tvar a zapracují ho do programu.



9. úkol. Úkolem je vytvořit obrázek, který se bude několikrát střídat se svým inverzním (v němž budou svítit ty diody, které jsou v původním obrázku zhasnuté). Každý obrázek se zobrazí třikrát, žáci tedy použijí opakování.

1. VÝPIS NA LED DISPLEJ

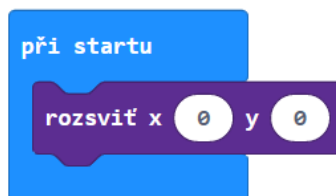
Ukázka řešení:



3. krok 15 minut



10. úkol. Žáci budou pracovat se souřadnicemi. Nejprve vyzkouší tento program:



Žáci poté mění čísla a pozorují, mohou přidat další blok rozsviť.

Poté promítněte žákům tento program a nechte žáky poradit se mezi sebou, co si myslí, že se na displeji zobrazí. Potom se zeptejte. Ptejte se vícekrát. Pokud se žáci neshodnou, neříkejte jim, co je správně. Pokud jedni druhé nepřesvědčí, nechte je program zapsat a vyzkoušet.



Program pro zadání úlohy

správná odpověď:



x – sloupec, zleva doprava od 0 do 4

y – řádek shora dolů od 0 do 4

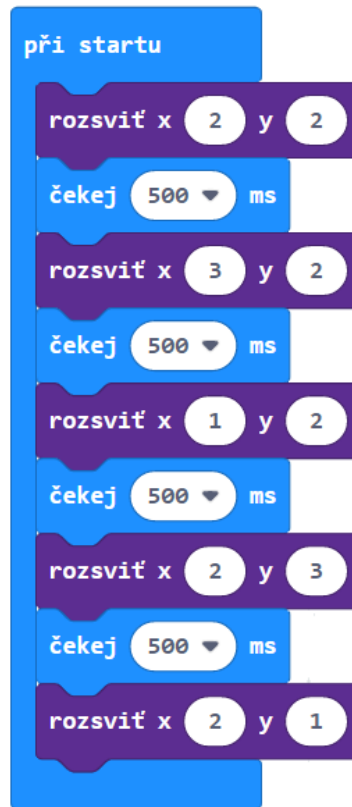
4. krok 10 minut



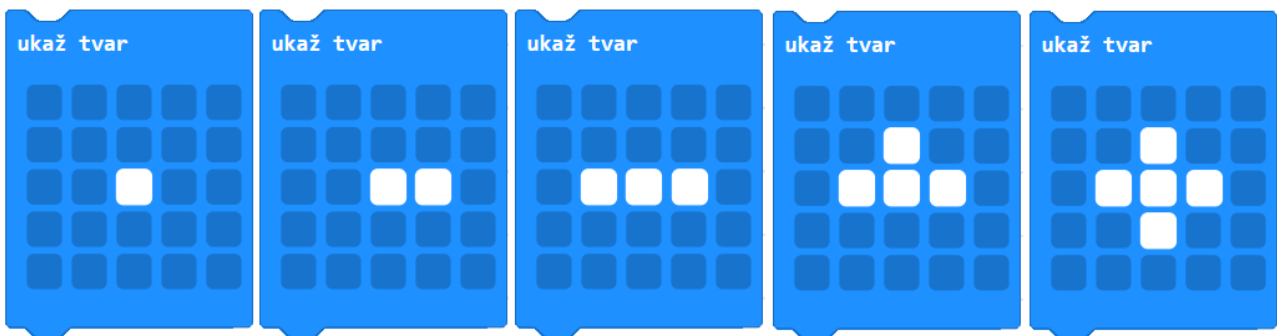
11. úkol. Žáci dostanou za úkol vykreslit na displeji svůj obrázek postupně.

Sestaví program z bloků rozsviť a čekej.

Příklad takového programu:



Slabší žáci, kteří mají potíže se souřadnicemi, mohou nakreslit postupně se objevující obrázek pomocí více bloků `ukáž tvar`.



Kreslit takové obrázky lze zadat i za dobrovolný domácí úkol. Žáci mají doma přístup k editoru se simulátorem, žáci mohou soubor s projektem uložit a ve škole nahrát do skutečného micro:bitu.

5. krok 20 minut



12. úkol. Nechejte žáky sestavit scénář, který z bodů **postupně** vykreslí první písmeno jejich jména.

Zkušenější žáci mohou využít cykly a proměnné. Písmeno se může vykreslovat stále dokola.

Ukázka řešení pro písmeno J:

The image shows a micro:bit simulator interface. On the left, the micro:bit board is displayed with its LED matrix showing the letter 'J' in red. The board has a green toggle switch at the top, two push buttons labeled 'A' and 'B', and a USB-C port. The bottom edge shows the pin headers labeled 0, 1, 2, 3V, and GND. On the right, the Scratch code is visible, starting with a 'při startu' (when started) block. The code consists of the following blocks:

- nastav číslo na 0** (set number to 0)
- opakuj 4 krát** (repeat 4 times) loop containing:
 - udělej** (do) block containing:
 - rozsviť x 3 y číslo** (turn on LED at x=3, y=number)
 - změň číslo o 1** (change number by 1)
 - čekej 100 ms** (wait 100 ms)
- rozsviť x 2 y 4** (turn on LED at x=2, y=4)
- čekej 100 ms** (wait 100 ms)
- rozsviť x 1 y 4** (turn on LED at x=1, y=4)
- čekej 100 ms** (wait 100 ms)
- rozsviť x 0 y 3** (turn on LED at x=0, y=3)



13. úkol. Vyzvěte žáky, aby prozkoumali blok zhasni. Poté nechejte žáky přijít na to, jak diody postupně zhasínat. Postačí využít simulátor.

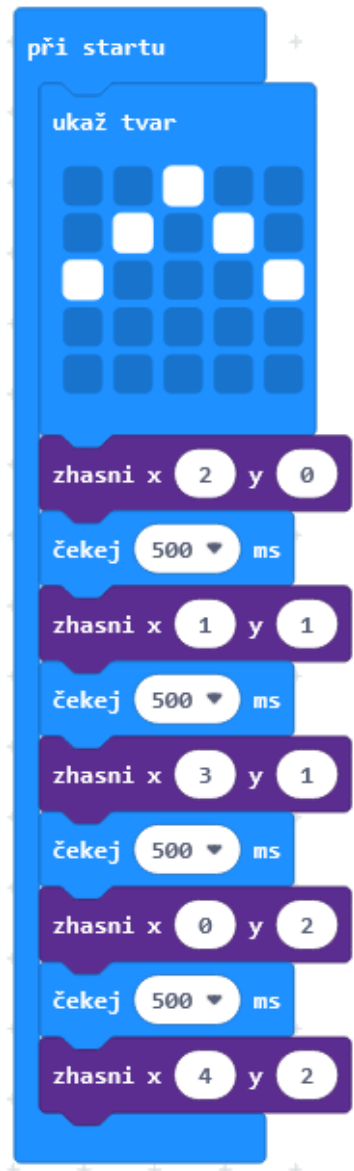
Aktivita má otevřený konec, na jejím konci je potřeba se zeptat žáků, co zjistili, požádat je, aby předvedli svůj program.

1. VÝPIS NA LED DISPLEJ



14. úkol. Žáci nechají vykreslit vlastní obrázek pomocí bloku `ukaz tvar` a ten pak by měl postupně po jednotlivých bodech zhasnout. Žáci mají přijít na to, které bloky pro postupné zhasínání použít.

Ukázka řešení:



Pokud máte čas, můžete si s žáky zahrát hru, ve které pracují ve dvojicích. Každý ze žáků zapíše na papír obrázek pomocí souřadnic, nebude jej kreslit. Poté si papírky vymění se spolužákem, který bude mít za úkol pomocí bloku `Ukaz tvar` obrázek vykreslit. Poté dvojice zkontroluje, jestli nakreslené obrázky odpovídají tomu, co žáci vymysleli.

PRŮVODCE HODINOU I-4, I-5

Žáci se seznámí s pokročilejší grafikou na micro:bitu. Naučí se tvorbě animace a zopakují souřadnice konkrétní diody displeje.

Rozšiřující látka s tvořivým potenciálem. Obsahuje množství úloh, na kterých žák procvičí samotné ovládání prostředí *Makecode*, skládání scénářů a ovládání micro:bitu. Důležitý je 6. krok – diskusní úloha.

Jakým způsobem a zda vůbec učit tuto část, ponecháváme na učitelích. Tuto hodinu lze rozdělit do dvou samostatných hodin nebo odučit jen prvních několik kroků. Je možné rychlejší žáky nechat pracovat samostatně napřed.

1. krok



15. úkol. Promítněte následující scénář v editoru a ukažte jeho animaci (doporučujeme spouštět tlačítkem, abyste měli jeho prezentaci pod kontrolou). Na displeji se ukáže animace startu rakety.

Vyzvěte žáky, aby nakreslili podobnou svoji animaci.



2. krok

Připomeňte žákům použití bloku rozsviř:

- x – sloupec, zleva doprava od 0 do 4
- y – řádek shora dolů od 0 do 4

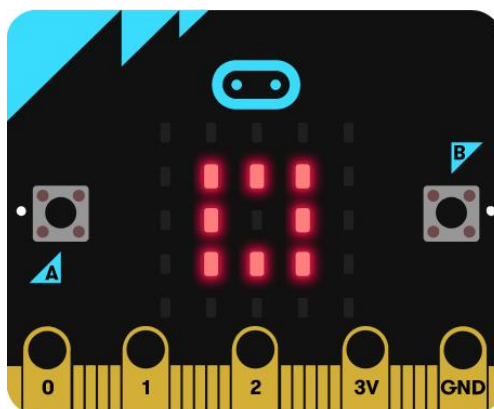
Bod vlevo nahoře je tedy 0, 0, bod vpravo dole 4, 4.



16. úkol. Ukažte žákům následující scénář. Nechejte je poradit se mezi sebou a přijít na to, jaký tvar se na displeji rozsvítí:



Správná odpověď:



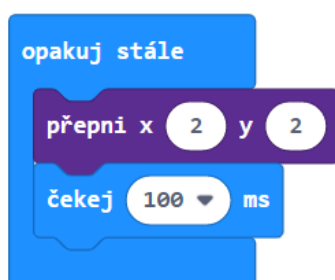
3. krok



17. úkol. Zadejte žákům, aby vyzkoušeli, co dělá blok `přepni`. Dejte jim čas na experimentování a potom se zeptejte, co blok dělá.

- Vyvolejte více žáků, aby svými slovy vyjádřili, co blok s diodami provádí. Žáci by měli vysvětlit, že blok dělá něco jiného, když je dioda rozsvícená a když je zhasnutá.
- Zeptejte se také žáků, jakým způsobem zjišťovali, co tento blok dělá. Popsat, jakým způsobem experimentuje, je pro žáka důležitá dovednost. Můžete se zeptat, jestli ke zkoumání někdo vyzkoušel blok opakuj.

Zeptejte se žáků, co bude dělat tento blok:



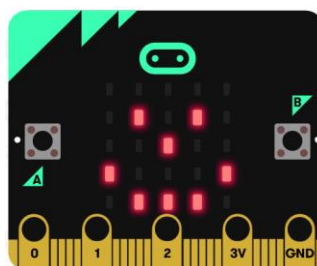
Žáci by nejprve měli odpovědět, poté sestaví scénář a vyzkouší, mění hodnoty v blocích `přepni` a `čkej`.

4. krok



18. úkol. Ukažte žákům „smutný“ micro:bit, vytvořený blokem `ukaz ikonu` (obr. vlevo). Vyzvěte je, aby **z něj** použitím bloků `přepni` vytvořili obrázek usmávajícího se smajlíka (obr. vpravo).

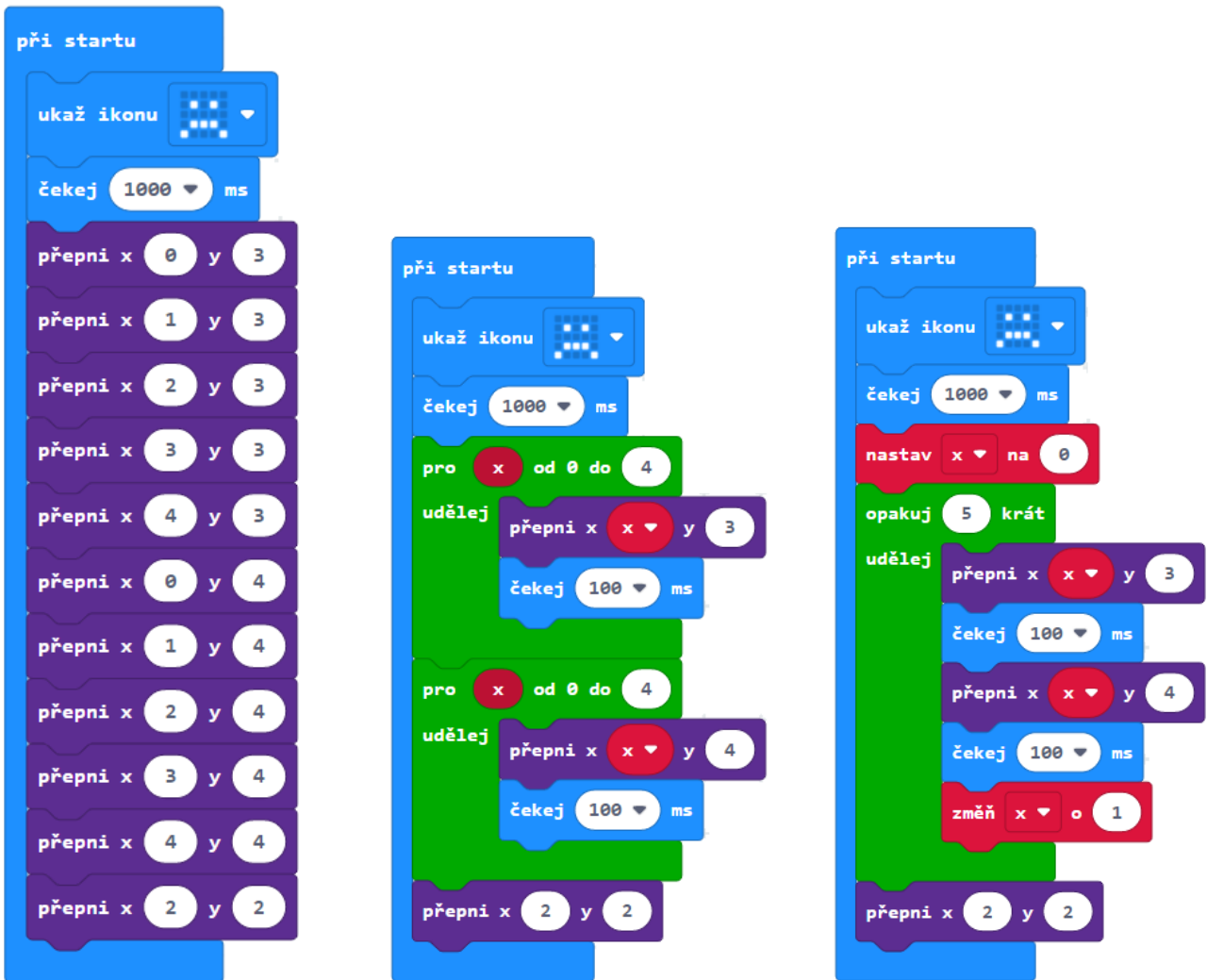
Obrázek jsme volili s „nosíkem“, protože tato ikona není v nabídce ikon.



Abyste se vyhnuli tomu, že někteří žáci budou tendenci si práci ulehčit tím, že obrázek vykreslí snadněji pomocí bloku `ukaz tvar` nebo `ukaz ikonu`, čtěte po nich, aby se jeden obrázek v druhý měnil postupně, s použitím bloku `čkej`.

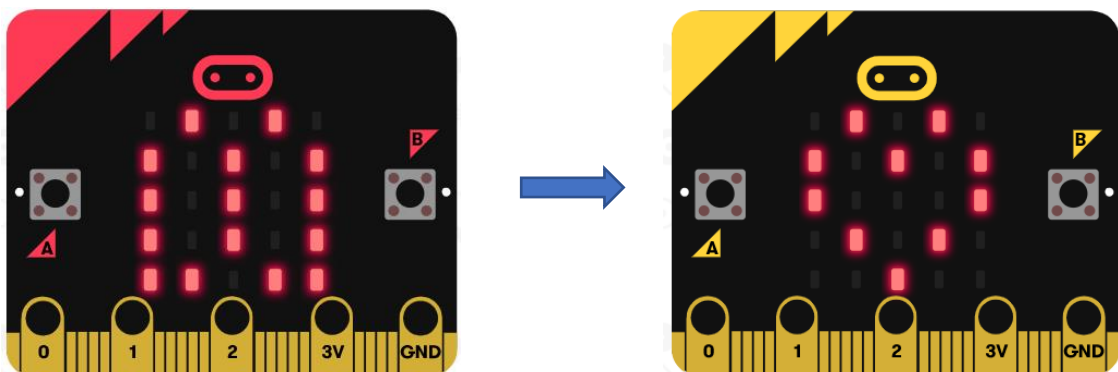
1. VÝPIS NA LED DISPLEJ

Možné správné řešení (vlevo jednoduché skládání bloků přepni, uprostřed a vpravo pokročilejší programování pomocí opakování a s doplněným čekáním):



Po naprogramování mohou žáci v úvodním bloku vyměnit ikonu za veselého smajlíka a pozorovat, co se stane. Mohou nastavit i jiné úvodní ikony.

Mohou také vyměnit první blok scénáře za ukať tvar a nakreslit takový tvar, který po provedení přepnutí podle scénáře vytvoří zajímavý nápad:



5. krok



19. úkol. Zadejte žákům následující úlohu: Honza vytvářel v micro:bitu obrázek šipky dolů, ale jeho scénář vytvořil šipku doprava. Žáci mají opravit tento scénář, který promítnete:

```

při startu
rozsviť x 0 y 2
rozsviť x 1 y 2
rozsviť x 2 y 2
rozsviť x 3 y 2
rozsviť x 3 y 1
rozsviť x 3 y 3
rozsviť x 4 y 2
  
```

Řešení:

```

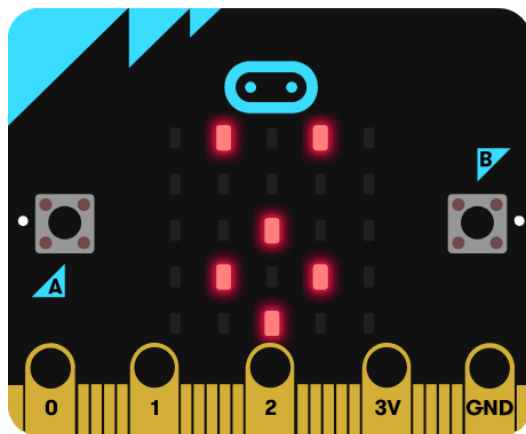
při startu
rozsviť x 2 y 0
rozsviť x 2 y 1
rozsviť x 2 y 2
rozsviť x 2 y 3
rozsviť x 1 y 3
rozsviť x 3 y 3
rozsviť x 2 y 4
  
```

6. krok

Aktivita bez počítače.



20. úkol. V této úloze mají žáci vybrat ze tří možností scénář, který vykreslí obrázek na displeji. Vyzvěte je, ať se poradí se sousedem a rozhodnou se pro jedno řešení. Na vyzvání by měli svůj výběr zdůvodnit.



a) při startu

- rozsviť x 0 y 3
- rozsviť x 0 y 1
- rozsviť x 2 y 2
- rozsviť x 3 y 1
- rozsviť x 3 y 3
- rozsviť x 4 y 2

b) při startu

- rozsviť x 3 y 0
- rozsviť x 1 y 0
- rozsviť x 2 y 2
- rozsviť x 1 y 3
- rozsviť x 3 y 3
- rozsviť x 2 y 4

c) při startu

- rozsviť x 2 y 0
- rozsviť x 4 y 0
- rozsviť x 3 y 2
- rozsviť x 4 y 3
- rozsviť x 2 y 3
- rozsviť x 3 y 4

Správná odpověď: b)

Odpověď a) má prohozené souřadnice x a y; vznikne tak stejný obrázek, ale „naležato“, překlopený podle úhlopříčky.

Odpověď c) je stejný obrázek, posunutý o 1 sloupec doprava.

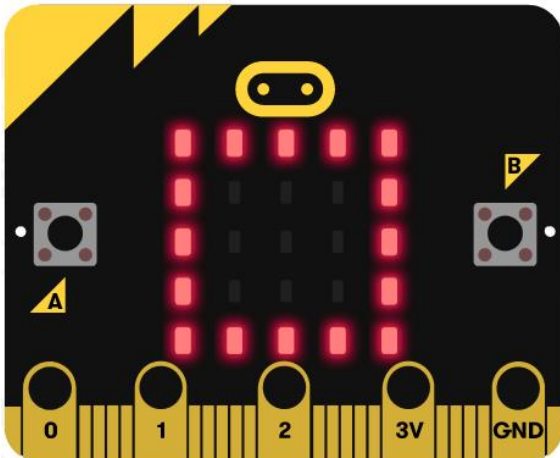
7. krok

Postupné rozsvěcení bodů pomocí cyklů. Úloha pro pokročilé žáky.



21. úkol. Vyzvěte žáky, aby vytvořili scénář, který postupně rozsvítí krajní diody s pauzou 200 ms. Měly by se rozsvěcet po směru hodinových ručiček. Žáci by měli použít bloky pro opakování.

Výsledný obrázek na displeji:



V řešení vpravo se rohové diody rozsvěčují „dvakrát“. Šikovnější žáky můžete nechat tento problém odstranit.

```

při startu
  nastav číslo na 0
  opakuj 5 krát
    udělej
      rozsviř x číslo y 0
      změň číslo o 1
      čekej 200 ms
  nastav číslo na 0
  opakuj 5 krát
    udělej
      rozsviř x 4 y číslo
      změň číslo o 1
      čekej 200 ms
  nastav číslo na 4
  opakuj 5 krát
    udělej
      rozsviř x číslo y 4
      změň číslo o -1
      čekej 200 ms
  nastav číslo na 4
  opakuj 5 krát
    udělej
      rozsviř x 0 y číslo
      změň číslo o -1
      čekej 200 ms
  
```


2 PRÁCE S TLAČÍTKY

Co se naučíte

- Ovládat obě programovatelná tlačítka
- Sestavovat programy reagující na stisk tlačítka

Co budete potřebovat

- PC s internetovým připojením
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

Časová náročnost

- 1 až 2 vyučovací hodiny po 45 minutách

PRŮVODCE HODINOU II-1

1. krok 10 minut

Na výzvu žáci odhalí, že micro:bit obsahuje celkem tři tlačítka. Dvě na přední straně (A, B) a jedno na zadní straně.

Tlačítko umístěné na zadní straně nás nebude v této lekci zajímat. Nelze jej programovat a slouží pro reset micro:bitu.

Programovatelná tlačítka A a B lze programovat pomocí bloků v sekci **Vstup**.



1. úkol. Začněte jednoduchým příkladem, nechejte žáky vyzkoušet tyto scénáře.



Žáci by měli objevit a říci, že micro:bit nebude dělat nic, dokud se jedno z tlačítek nezmáčkne.

Vyzvěte žáky, aby dovnitř bloků umístili i jiné bloky, například `zobraz text`.

2. krok 5 minut



2. úkol. Vyzvěte žáky, aby v rozbalené nabídce bloku `po stisknutí tlačítka` vybrali třetí možnost **A+B**, přidali třetí blok a vyzkoušeli, jak spustit tento třetí blok.

Žáci by měli vytvořit takovýto scénář:



Bloky umístěné uvnitř tohoto bloku se provedou jen tehdy, pokud budou zmáčkuta obě tlačítka současně.

Zeptejte se žáků, k čemu se tento blok hodí.

Simuluje další tlačítko.

3. krok 15 minut



3. úkol. Zaúkolujte žáky, aby prozkoumali blok při zatřesení. Scénář přidají k ostatním scénářům – nabídka různých akcí se bude zvětšovat.

Žáci by měli vytvořit takovýto scénář:



Stisknutí tlačítka nebo zatřesení říkáme události (které spouštějí akci, v tomto případě rozsvícení displeje). Událostmi se více zabývá kapitola 4.

Zeptejte se žáků, k čemu tyto události využít.

Při vytváření her, při fyzikálních měřeních, při spouštění animací a podobně.

4. krok 15 minut



4. úkol. Vyzkoušíme použití tlačítka pro ovládání počítadla. Žáci dostanou za úkol sestavit program, který bude neustále na displeji zobrazovat, kolikrát bylo stisknuto tlačítko A.

Žáci sestaví takovýto program (ukazujeme dvě varianty, které mohou sestavit)



Žáci si vytvoří proměnnou, která si bude pamatovat, kolikrát se tlačítko stisklo. Program její hodnotu bude stále zobrazovat na displeji a při stisknutí tlačítka A se její hodnota zvětší o 1. Při startu je třeba proměnnou vynulovat.

Návod z předchozího odstavce nemusíte žákům říci; cennější je, když jej žáci společně „vymyslí“. K tomu použijte návodné otázky:

1. Jak si bude micro:bit pamatovat, kolikrát se tlačítko stisklo?
2. Jak zařídit, aby na displeji svítila zapamatovaná hodnota?
3. Jak naprogramovat, co se stane po stisknutí tlačítka?

Dále nechejte žáky přidat scénář, který po stisknutí tlačítka B naopak jedničku odečte.

Žáci si vzpomenou, že odečítání znamená změnit hodnotu proměnné o -1.

Zeptejte se žáků, k čemu by se dal takto naprogramovaný micro:bit použít.

Může sloužit při měření v terénu k počítání lidí, kteří projdou po chodníku, aut, která projedou, ptáků na rybníce apod. Tlačítkem B pak můžeme opravit chybu, kdybychom se „ukliknuli“.

Při pandemii můžeme počítat aktuální počet osob v obchodě: tlačítkem A přidáváme příchozí a tlačítkem B odcházející zákazníky.

5. krok - dobrovolné

Rozšiřující úloha pro žáky, kteří již umí dobře programovat. Zabere podstatnou část vyučovací hodiny.



5. úkol. Zadejte žákům samostatnou práci ve dvojici, naprogramovat postřehovou hru **Chytání písmenek**. Na micro:bitu se bude střídavě náhodně zobrazovat a zhasínat písmeno A nebo B a hráč bude muset stisknout odpovídající tlačítko, než písmeno zmizí z displeje. Za „chycené“ písmenko se hráči přičte bod.

Hra skončí po daném počtu opakování. Dobu zobrazení a čekání na stisk lze při ladění hry nastavit tak, aby se hra dobře hrála (ani pomalé, ani rychlé). Na závěr může být vyhodnocení např. počtem úspěšných pokusů.

Pro volbu A nebo B žáci použijí blok náhodné číslo ze sekce **Matematika**. Budou potřebovat dvě proměnné (pro uchování hodnoty zobrazeného tlačítka a pro uchování počtu bodů).

Hru lze variovat:

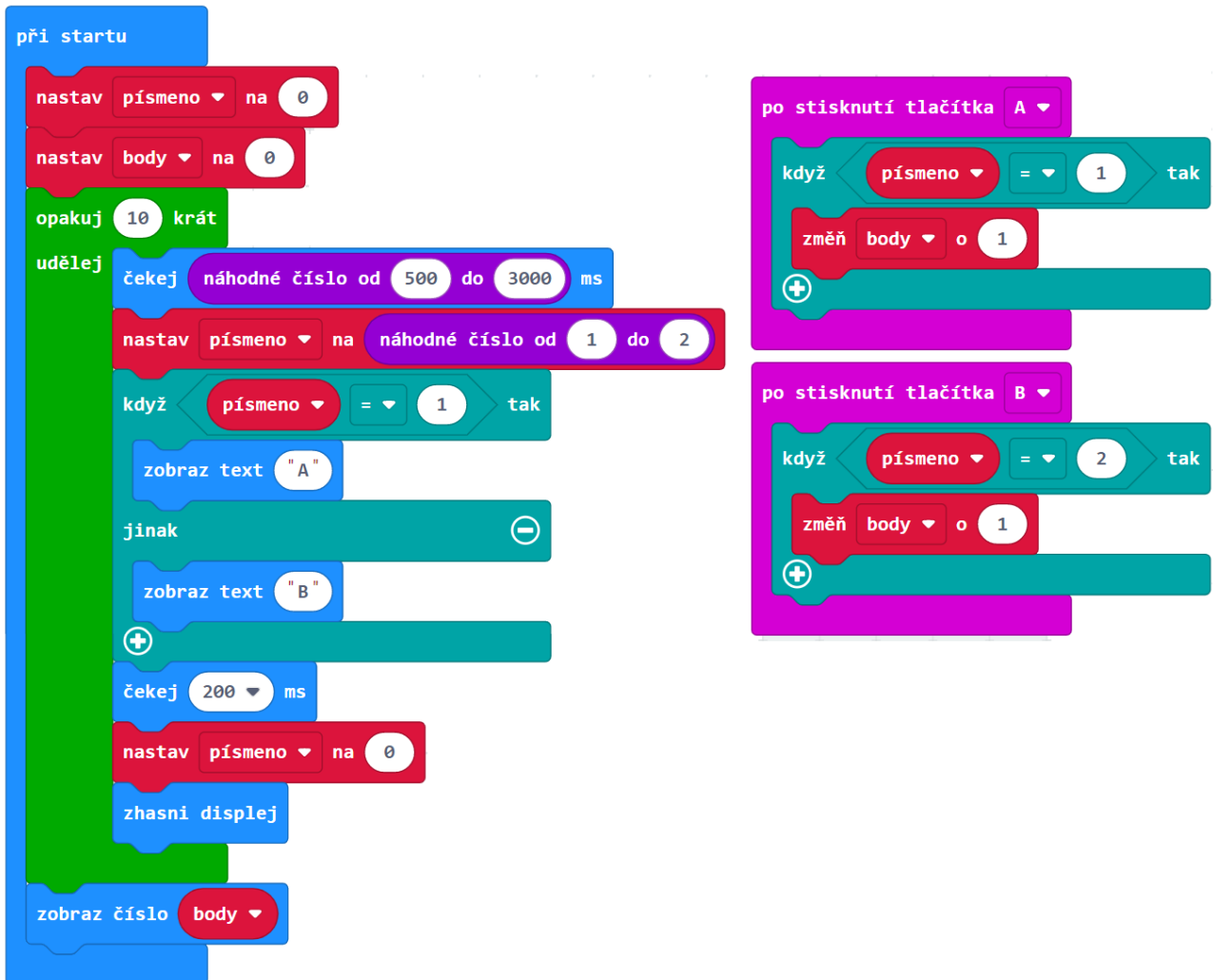
- Zobrazováním obrázků namísto písmen, kdy si hráč musí pamatovat, ke kterému tlačítku přísluší který obrázek.
- Ukončením hry po stisknutí obou tlačítek A, B současně
- Při stisknutí nesprávného tlačítka nebo stisknutí v nesprávnou dobu se bod odečte

Vysvětlení vzorového řešení na další stránce: hra má deset kol. V každém kole se nejprve čeká 0,5 až 3 sekundy. Poté se náhodně vybere číslo 1 nebo 2, které odpovídá tlačítku A nebo B. Program pak čeká 0,2 s, po které displej písmena zobrazuje a je možné je „chytit“. Poté písmeno z displeje zmizí a hodnota proměnné *písmeno* se vynuluje.

2 PRÁCE S TLAČÍTKY

Hodnota v proměnné *písmeno* ukazuje, které tlačítko je zvoleno a zda je zobrazeno (1 = písmeno A, 2 – písmeno B, 0 = nic není zobrazeno). Po každém stisku tlačítka se kontroluje, jestli je písmeno zobrazeno na displeji a zda je to správné číslo. Pak se přičte bod. Na konci hry se body zobrazí.

Ukázka možného řešení:



Pokud postupujete přímo podle učebnice, budete na příští hodinu potřebovat ke každému micro:bitu dva vodiče s krokodýly a nějaký hardware pro zvukový výstup.

Řekněte žákům ať si přinesou na příští hodinu sluchátka nebo repráček s jackem. Případně si připravte dostatečný počet akustických bzučáků.

Doporučujeme, aby žáci měli sluchátka, ať se vzájemně neruší. Vy si naopak připravte repráčky, ať vše můžete dobře demonstrovat.

Micro:bit 2: Pokud používáte micro:bit 2, nemusíte si připravovat další hardware, neboť micro:bit 2 má už zabudovaný vlastní reproduktor. Nicméně z důvodu eliminace hluku doporučujeme žákům sluchátka.

3 HUDBA

Co se naučíte

- Připojit k micro:bitu reproduktor, bzučák nebo sluchátka
- Přehrát přednastavený zvuk
- Vytvořit vlastní melodii

Co budete potřebovat

- PC s internetovým připojením
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reproduktor nebo sluchátka s konektorem jack, popřípadě akustický bzučák (piezzo buzzer).

Časová náročnost

- 2 vyučovací hodiny po 45 minutách

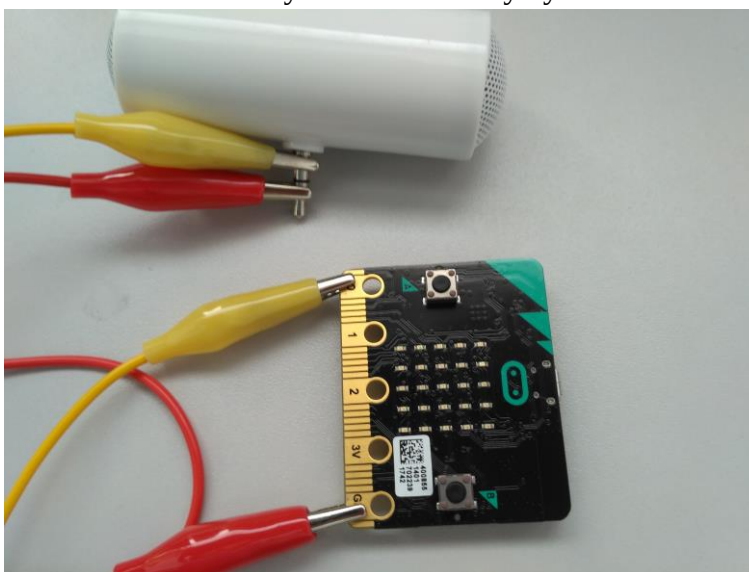
PRŮVODCE HODINOU III-1

Žáci si připojí k micro:bitu hardware pro přehrání zvuku a ozvučí tak své projekty.

1. krok 15 minut

Rozdejte žákům micro:bity a kabely. Řekněte jim, ať si připraví sluchátka. Raději mějte připravená sluchátka pro ty, kteří si je zapomenou. Vysvětlete žákům zapojení.

Micro:bit verze 1 nemá přímý audio výstup, ale připojení externího reproduktoru je velmi snadné. Budete potřebovat dva vodiče nejlépe opatřené na koncích krokodýlky. Jeden na dolní straně micro:bitu připněte k pinu GND a druhý k pinu 0. Tak praví manuál. Na základě osobních zkušeností můžeme potvrdit, že první vodič obvykle může být i na 1 anebo na 2 a dokonce pak někdy může být zvuk lepší – záleží na použitém zařízení. Druhý vodič musí vždy být na GND. Opačné konce vodičů připojte k jack konektoru libovolného reproduktoru či sluchátek. Nezáleží na pořadí, který vodič kam připojíte. Má-li váš jack tři vstupy, pak jeden z vodičů připojte na prostřední a druhý na libovolný z krajních vodičů. Má-li čtyři vstupy, pak by měly fungovat buď oba krajní nebo oba vnitřní (možná budete muset trochu experimentovat). Také můžete použít jako výstup bzučák, pak prostě připojíte každý vodič k jednomu z pinů. Viz obrázek.



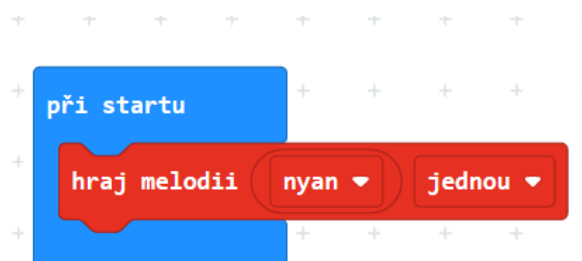
Počítejte s tím, že v případě kvalitních reproduktorů může být výstup poměrně hlasitý a nastavte výstup na nižší úroveň.

Micro:bit 2: Pokud ve třídě používáte micro:bit verze 2, tento krok můžete přeskočit. Tato verze micro:bitu obsahuje vlastní reproduktor, nemusíte tedy nic zapojovat a hned se můžete pustit do programování. Dejte si pozor, že tento reproduktor nejde nijak hardwareově ztlumit ani odpojit – po zapnutí micro:bitu bude hrát do té doby, než nahrajete jiný program.

2. krok 15 minut



1. úkol. Vyzkoušejte přehrávání na připravené melodii. Nahrajte do micro:bitu následující program:



Tento blok přehraje jednu z připravených melodií jednou, ve smyčce. Odlaďte u všech žáků přehrávání hudby. Pokud někomu hudba nehraje, zkuste jiný hardware. Tato melodie je vhodná pro testování – je dlouhá a výrazná.

Nechte žáky blok vyzkoušet a prozkoumat. Nakonec se zeptejte:

Jaký rozdíl je mezi možnostmi jednou/stále dokola a jednou/stále dokola na pozadí?

V první možnosti micro:bit čeká, až se melodie přehraje celá a pak pokračuje ve vykonávání programu, ve druhém případě spustí melodii a pokračuje v kódu dál.

3. krok 15 minut

Nyní se zkombinuje vše, co již žáci znají. Zobrazení obrázku, práce s tlačítky a přehrání melodie.



2. úkol. Žáci mají za úkol sestavit takový program, který po zmáčknutí tlačítka A zobrazí na displeji jednu z předpřipravených ikon a přehraje nějakou melodii. Obdobně u tlačítka B.

Možné řešení:



PRŮVODCE HODINOU III-2

Žáci si na micro:bitu připraví vlastní melodii.

Co bude v této hodině potřeba:

- PC s internetovým připojením
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukční nebo sluchátka s jackem, popřípadě piezzo buzzer.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro žáky

1. krok 10 minut

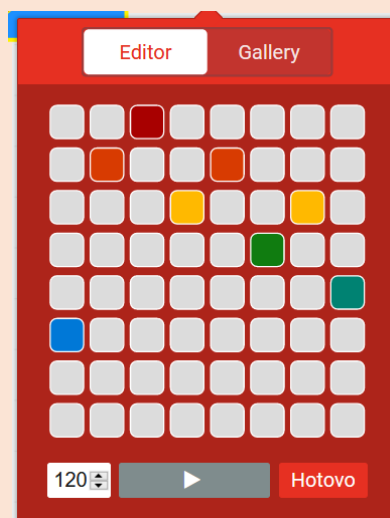
Rozdejte žákům micro:bity a kabely. Nechte je připojit sluchátka.



3. úkol. Nechejte žáky prozkoumat blok přehraj melodii:



Tento blok dovolí žákům vytvořit vlastní melodii. Klepnutím na noty se otevře editor not, ve kterém si mohou poskládat vlastní melodii. Melodie použitá v příkladu se dá nalézt v Galerii pod názvem *Mystery*.



Nechejte žáky, aby si vytvořili vlastní melodie. Je to cenná tvůrčí činnost. Žáci mohou blok několikrát zkopírovat a vytvářet tak delší melodie.

2. krok 15 minut



4. úkol. Hudba může sloužit také jako událost, podmínka pro spuštění scénáře. Žáci přidají do programu následující dva scénáře a prozkoumají, co dělají.



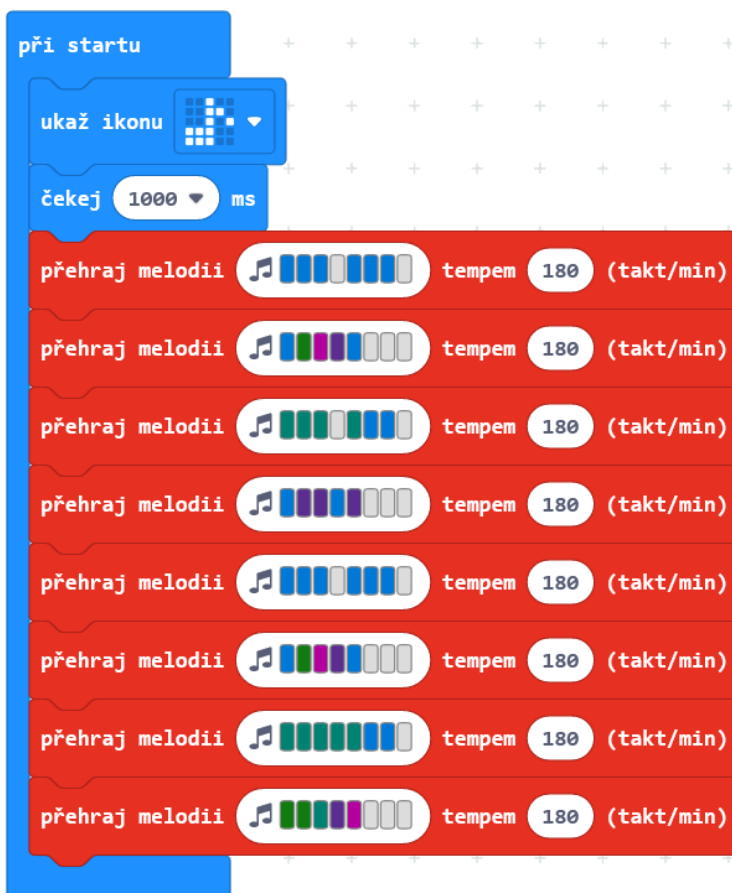
První scénář způsobí, že se micro:bit bude usmívat od chvíle, co začne hrát hudba. Druhý mu pak nastaví smutný výraz poté, co melodie skončí.

Nechte žáky, aby si vyzkoušeli všechny možné podmínky z tohoto bloku.

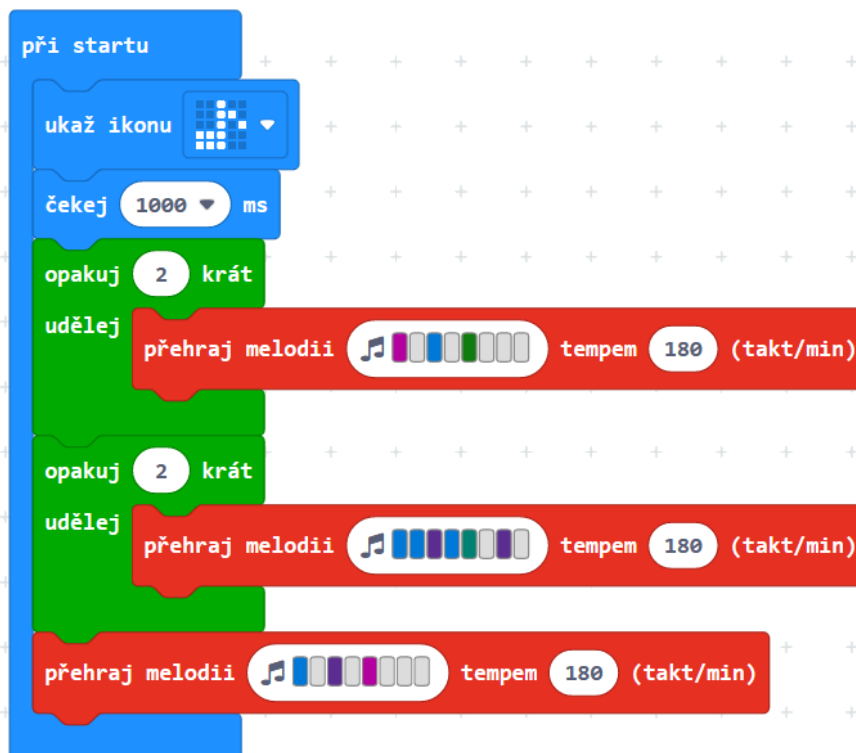
3. krok 20 minut



5. úkol. Vyzvěte žáky, ať si sestaví vlastní melodii nebo ať naprogramují přehrání nějaké známé melodie. Například takto může vypadat scénář ke koledě *Rolničky*:



Jiný příklad: 1. sloka písničky *Ovčáci čtveráci*:



4 POLOHA

Co se naučíte

- Pracovat s orientací a pohybem micro:bitu v prostoru
- Pracovat s vestavěným akcelerometrem, pochopit jeho možnosti a využít jej
- Používat události
- Pracovat s vestavěným kompasem
- Používat detekci intenzity světla

Co budete potřebovat

- PC s přístupem k internetu
- Propojovací USB kabel s micro USB konektorem
- Micro:bit

Časová náročnost

- 4 vyučovací hodiny po 45 minutách

PRŮVODCE HODINOU IV-1

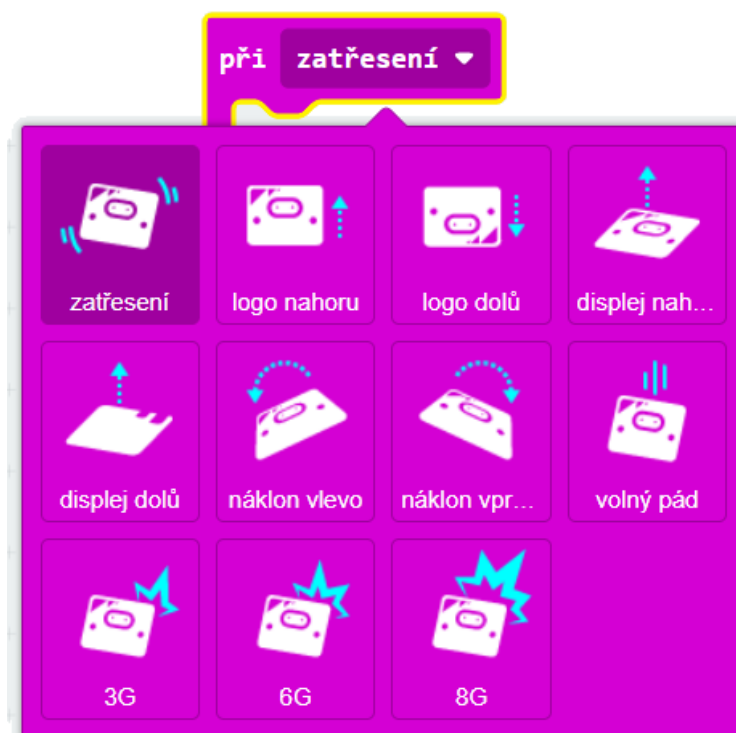
Žáci se v této hodině naučí, jakým způsobem zjistit orientaci micro:bitu v prostoru a jak tuto informaci využít.

1. krok 30 minut

Rozdejte žákům micro:bity a kabely.



1. úkol. Žáci si prohlédnou blok při zatřesení ze sekce **Vstup** – rozbalí jeho nabídku, prohlédnou si a na dotaz učitele zkusí odhadnout, co jednotlivé události v této nabídce znamenají.



Události 3G, 5G, 8G vysvětlujeme pouze, pokud se některý žák zeptá – pro naše úlohy je nepotřebujeme. Tyto události znamenají velikost přetížení. Lze jej vysvětlit tím, „jak hodně se zrychlí nebo jak rychle se zastaví pohyb micro:bitu“ (a průměrem, že volný pád znamená 1G). Přetížení mohou žáci znát z kosmonautiky nebo akrobatického létání.

Diskutujte se žáky pojem akcelerometr (zařízení pro zjištění aktuální orientace, směru pohybu, zrychlení, popř. volného pádu nebo zrychlení). Tento senzor pozná, jak je micro:bit nakloněn, jestli se s ním zatřepalo nebo zda padá ze stolu, a program na to může reagovat.

Zeptejte se žáků na využití akcelerometru – herní ovladač, generátor událostí.



2. úkol. Zadejte žákům vyzkoušet náklon micro:bitu vlevo a vpravo jako událost pro spuštění. Žáci sestaví program, který na náklon micro:bitu zobrazí na displeji různé ikony.

Možné řešení:



3. úkol. Žáci vytvoří program, který po naklonění micro:bitu vlevo nebo vpravo ukáže šipku, která bude ukazovat nahoru, ke stropu učebny.

Vyzkoušením žáci zjistí, kterou šipku použít. Když je micro:bit nakloněn vlevo, nahoru bude ukazovat šipka vpravo, tedy „na východ“. Obdobně šipka „na západ“ ukazuje nahoru při opačném naklonění.

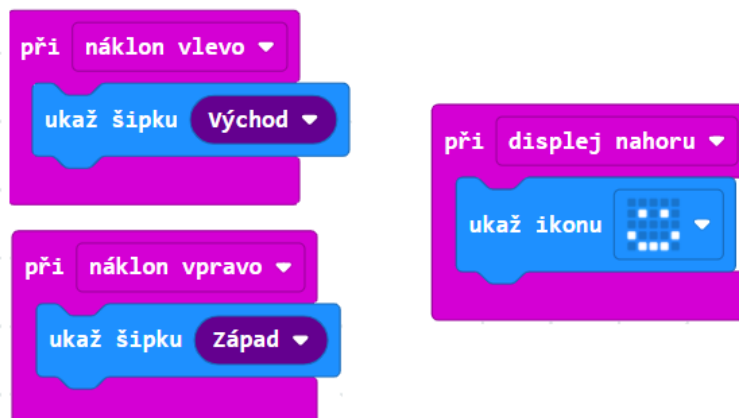
Ukázka řešení:



4. úkol. Předchozí program má nevýhodu, že šipka zůstává na displeji i tehdy, když se micro:bit vrátí do vodorovné polohy. Žáci mají za úkol přidat událost, která na displeji zobrazí smajlíka, když leží micro:bit vodorovně na stole.

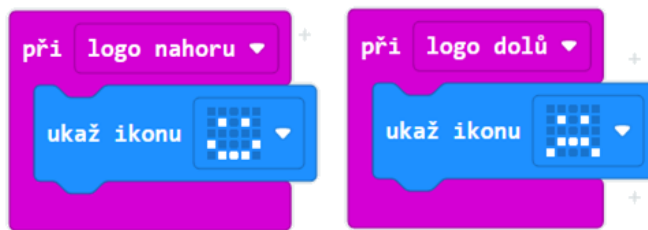
Žáky možná napadne, že vlastně vytvořili automatickou vodováhu, která buď ukazuje, že je předmět vodorovně, nebo šipkou ukazuje na jeho zvednutou část.

Ukázka řešení:



5. úkol. Zadejte žákům úkol vytvořit program, který zobrazí veselý smajlík, když se micro:bit postaví na stůl logem nahoru (tedy dírkami dolů), a smutný smajlík, když se převrátí vzhůru nohama.

Takto vypadá vzorový program:



2. krok 15 minut

V tomto kroku budou žáci sami vymýšlet použití události k něčemu užitečnému. Vymyšlený nápad by měli naprogramovat. Tvořivá aktivita.

Nechejte žáky vyzkoušet další události. Pokud budou testovat volný pád, měli by si dát pozor, aby micro:bit nerozbili.



6. úkol. Pro jednu z událostí by žáci měli vymyslet její využití a naprogramovat ji k tomu.

Poté se žáků zeptejte, co vymysleli a jak by se to v praxi použilo.

PRŮVODCE HODINOU IV-2

V této hodině žáci prohloubí svoje dovednosti práce s událostmi.

1. krok – 30 minut



7. úkol. Zadejte žákům vyzkoušet tento program.



Žáci by měli přemýšlet, k čemu by se takový program mohl hodit. Mohou odpovědět, kdyby tam byla čísla od 1 do 6, mohl by sloužit pro házení kostkou, pro čísla 0, 1 pro házení mincí. Na tomto příkladu ale vybírá ze tří možností, mohlo by tedy jít o hru *Kámen, nůžky, papír*.



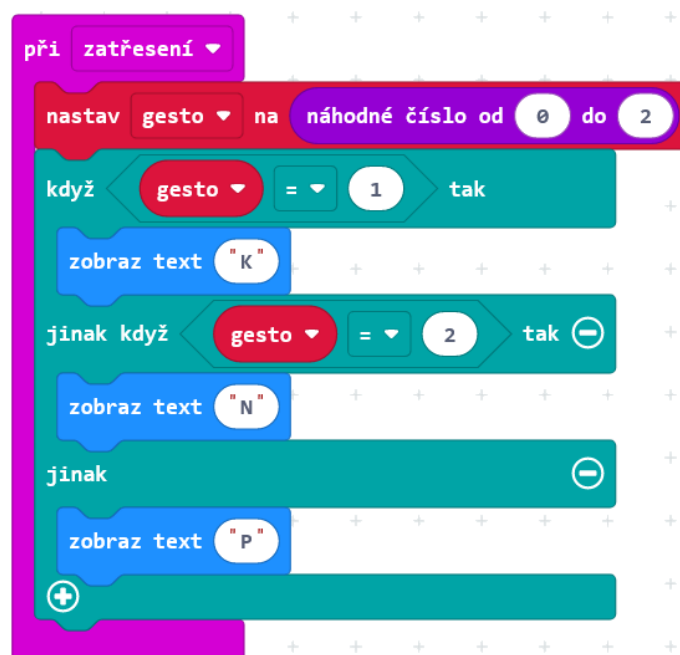
8. úkol. Nyní žáci vytvoří variaci hry *Kámen, nůžky, papír*. Micro:bit bude při zatřesení náhodně zobrazovat písmena K pro volbu kámen, N pro nůžky a P pro papír.

Protože micro:bit umí náhodně generovat pouze čísla, budeme muset písmena dodatečně přiřazovat jednotlivým volbám. K tomu potřebujeme si volbu nejprve zapamatovat, použít proměnnou.

Pro rozhodování budeme potřebovat blok *když* – pro rozhodování ze tří možností použijeme tlačítko ⊕ na bloku, které blok rozvětví.

Dejte žákům dostatek času k sestavení a vyzkoušení programu.

Příklad správného řešení:



Žáci mohou namísto písmen K, N, P nechat zobrazovat symboly.



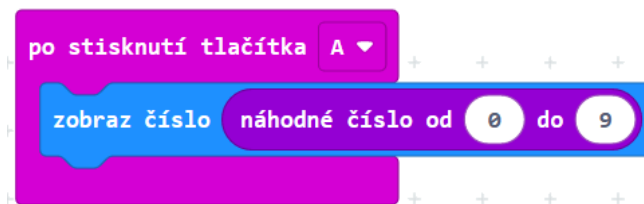
Každý žák si pak může zahrát sám se sebou. Jednou rukou hraje sám (udělá gesto kámen, nůžky nebo papír), v druhé ruce současně zatřese micro:bitem, a podívá se, jestli vyhrál on, nebo micro:bit. Může si pak s Micro:bitem zahrát zápas na tři vítězství.

2. krok – 15 minut



9. úkol. Nyní žákům zadejte následující úlohu: Vytvořte program, který po zmáčknutí tlačítka A napíše náhodné číslo od 0 do 9.

Ukázka řešení:

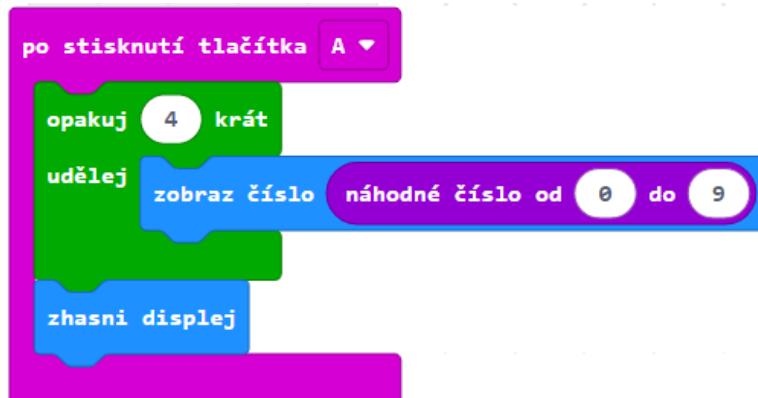


S hotovým programem si žáci mohou zahrát „loterii“. Nejprve si tipnou, jaké číslo padne, poté stisknou tlačítko. Mohou počítat, kolikrát se trefoří na 10 pokusů.



10. úkol. Použitím opakování mohou žáci generovat např. náhodné telefonní číslo, PIN nebo PSČ.

Možné řešení pro PIN:



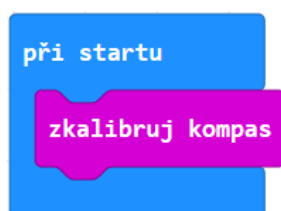
PRŮVODCE HODINOU IV-3

V této hodině se žáci naučí pracovat s micro:bitem jako s kompasem a stanovit pomocí něj azimut.

1. krok – 20 minut

Proberte se žáky pojmy kompas a azimut. Zeptejte se, zda těmto pojmům rozumí. Ukažte si ve třídě co nejpřesněji, kterým směrem je sever a ostatní světové strany. Vhodní je půjčit si z kabinetu zeměpisu jeden kompas.

Micro:bit obsahuje zabudovaný kompas, který současně lze použít jako čidlo intenzity magnetického pole. Tento kompas je nutné vždy před použitím kalibrovat, jinak nelze ručit za jeho správnou funkci. Kalibrace se provede pomocí následujícího bloku:



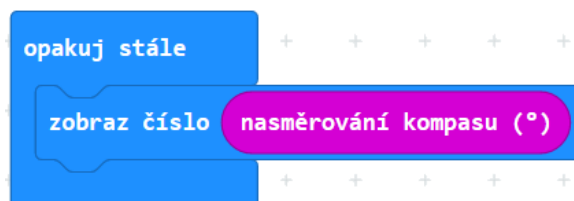
Kalibraci provedeme následovně: Na micro:bitu se před kalibrací zobrazí instrukce, jak postupovat. Text instrukce bohužel nelze přeskočit a je potřeba čekat, až doběhne úplně do konce. Poté je nutno naklánět a otáčet s micro:bitem tak dlouho, dokud celý displej nerozsvítíme. Může to trvat i přes minutu. Poté se asi po 5 sekundách rozsvítí smajlík – micro:bit je zkalibrován.

Tuto kalibraci je nutné provést před každým použitím kompasu. Zdůrazněte to žákům.

Pokud některý micro:bit ukazuje něco jiného než ostatní nebo než to, co očekáváte, stiskněte na něm tlačítko reset a opakujte kalibraci.



11. úkol. Žáci vyzkoumají, co dělá blok nasměrování kompasu. Sestaví k tomu tento scénář:



Nechejte žáky, ať se s micro:bitem otáčí a sledují čísla na displeji, ať zkusí nastavit na displeji co nejmenší číslo.

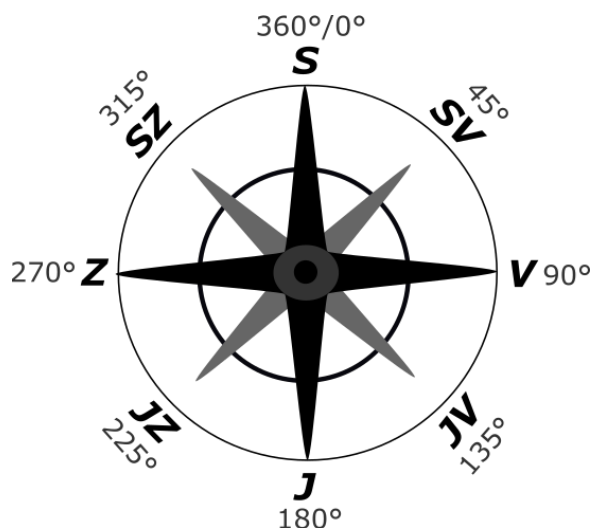
Žákům, kteří pracovali ve Scratchi, mohou čísla připomínat směr natočení postavy. Nebudou daleko od pravdy.

4 POLOHA

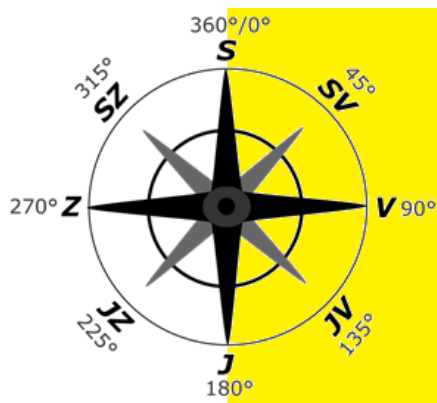
Micro:bit bude ukazovat tzv. azimut, tedy o kolik stupňů od severu je micro:bit odkloněn. Stupně přibývají po směru hodinových ručiček v intervalu od 0 do 360 (podobně jako směr postavy ve Scratchi vzhledem k směru *nahoru*).



12. úkol. Následující obrázek ukazuje hodnoty čísel, které kompas micro:bitu udává pro světové strany.



Obrázek s žáky projděte a zkuste s nimi přijít na podmínku, jaká bude platit tehdy, když micro:bit bude ukazovat více na východ než na západ (když bude ukazovat do pravé poloviny růžice světových stran – na obrázku žlutě vybarveno).



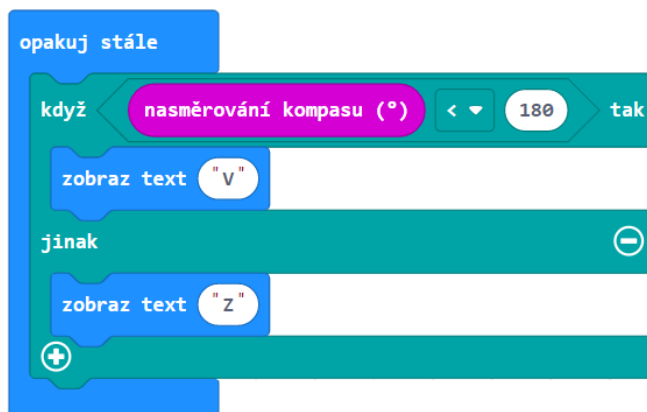
Ta podmínka zní, že hodnota nasměrování kompasu bude menší než 180°

2. krok – 10 minut



13. úkol. Zadejte následující úlohu: Honza chce vytvořit program, který by na micro:bitu písmeny V, Z ukazoval, jestli jde Honza na západ nebo na východ.

Ukázka správného řešení:



3. krok – 15 minut

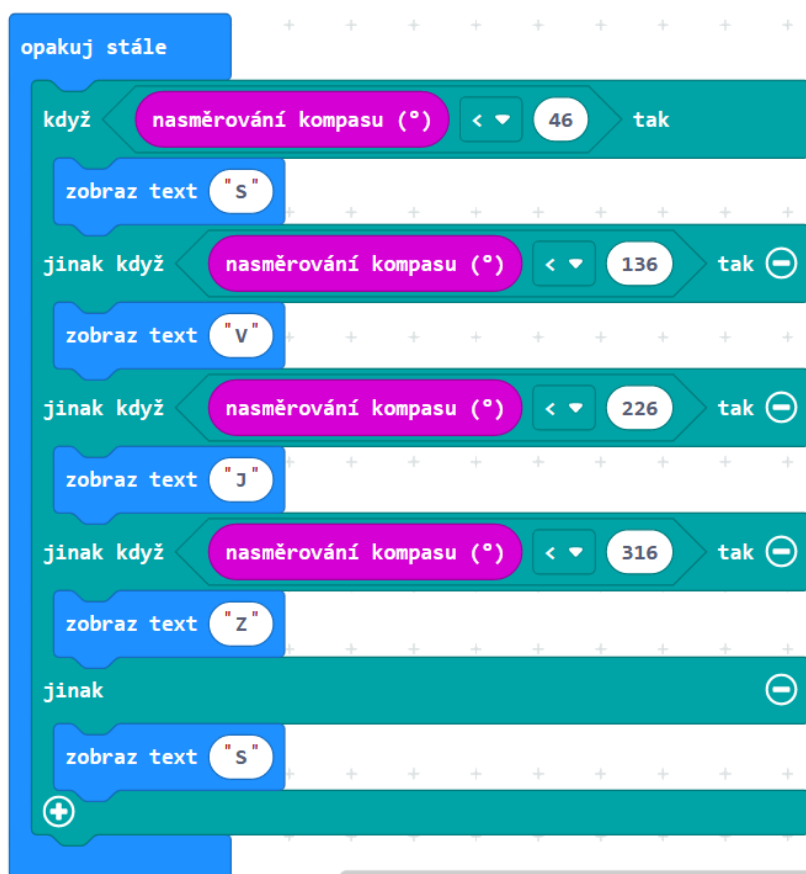


14. úkol. Úloha pro pokročilé nebo rychlé žáky. V této úloze program z předchozí úlohy žáci upraví tak, aby micro:bit ukazoval do čtyř světových stran S, V, J, Z.

V úvodní diskusi by žáci měli přijít na to, že za hranici mezi světovými stranami budeme považovat úhlopříčky. za východ pak bude micro:bit považovat intervaly úhlů (46, 135), za jih (136, 225) a za západ (226, 315) a za sever zbytek, tedy intervaly (0,45) a (316, 359),

Protože oblast severu je rozdělena na dva intervaly, máme pět částí, do kterých se všechny směry dělí. Musíme se tedy ptát pomocí složitějšího příkazu `když` se čtyřmi podmínkami.

Možné řešení:



PRŮVODCE HODINOU IV-4

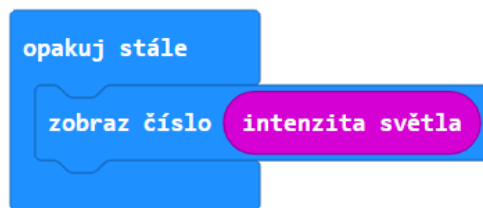
V této hodině se žáci naučí pomocí micro:bitu měřit intenzitu osvětlení.

1. krok – 20 minut



15. úkol. Nechejme žáky prozkoumat, co dělá blok intenzita světla ze sekce *Vstupy*. Dejte žákům za úkol vytvořit program, v němž bude displej stále zobrazovat hodnotu intenzity světla. Protože intenzita světla se udává v číslech, je třeba použít blok zobraz číslo.

Řešení:



Žáci natáčejí micro:bit k oknu, přikrývají rukou, svítí na něj mobilem apod. a pozorují, jaké hodnoty micro:bit ukazuje.

Žáci by vyzkoumat, jaká je největší a nejnižší hodnota, která se jim zobrazila. (Je to 0 až 255, možná krajních hodnot nedosáhnou, ale to nevádí – lze diskutovat, proč).

Žáci mohou přinášet k diskusi plodné myšlenky, např. že se nuly nedocílí ani v temné komoře, protože micro:bit sám svítí. Musel by nejprve zhasnout, naměřit a teprve poté zobrazit.

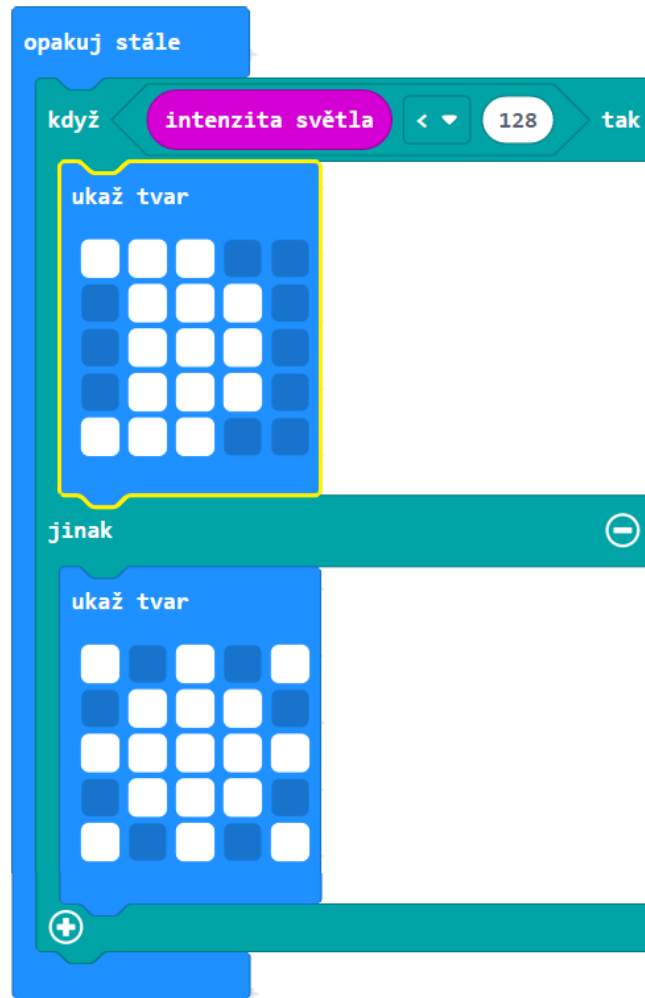
2. krok – 25 minut



16. úkol. Žáci vytvoří program, který rozezná noc a den. Když micro:bit rozhodne, že je intenzita světla malá, zobrazí ikonku měsíce, jinak zobrazí sluníčko.

Žáci po sestavení programu mohou experimentovat, hledat vhodnou hodnotu hranice mezi dnem a nocí a program dodatečně upravit. Hraniční hodnota může být i výrazně nižší nebo vyšší než prostřední hodnota mezi minimální a maximální hodnotou, tedy než 128 – záleží na světelných podmínkách ve třídě.

Příklady řešení:



5 SÍŤ

Tato kapitola se věnuje propojení dvou nebo více micro:bitů pomocí sítě.

Co se naučíte

- Bezdrátově propojit dva micro:bity a přenést informaci
- Používat příkazy na přenos dat pomocí rádiového spojení

Co budete potřebovat

- PC s internetovým připojením
- Propojovací USB kabel micro USB koncovkou
- Dva micro:bity do každé pracovní skupiny

Časová náročnost

- 5 vyučovacích hodin po 45 minutách

PRŮVODCE HODINOU V-1

Žáci se v této hodině seznámí s některými pojmy z počítačových sítí. Propojí si micro:bity a vyzkouší si radiový přenos signálu mezi nimi.

1. krok 15 minut

Žáci budou pracovat ve skupinách, v každé skupině budou dva micro:bity, jeden bude vysílat a druhý přijímat signál.

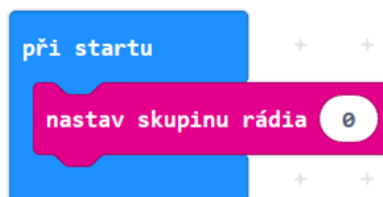
Je-li ve třídě dostatek micro:bitů pro každého žáka, budou pracovat ve dvojicích.

Téma můžete zahájit debatou se žáky o pojmech počítačová síť, drátová, bezdrátová. O bezpečnosti sítí se můžete zmínit po první úloze. Diskutujte o příkladech.

Micro:bity budou ke komunikaci používat radiový přenos. Můžete jej žákům připodobnit k vysílání rozhlasu – ten vysílá na různých kanálech a na ty je třeba radiopřijímač naladit. Micro:bity mohou používat 256 různých kanálů označených 0 až 255.



1. úkol. Aby si různé skupiny ve třídě navzájem nerušily vysílání, přiřadte každé dvojici jiné číslo skupiny. Do obou micro:bitů ve skupině, *přijímače* i *vysílače*, je třeba nejprve zadat skupinu (vysílací kanál) pomocí bloku



Číslo žáci zadají podle přiděleného čísla skupiny.

Zkoušku spojení žáci provedou zadáním bloků pro vysílání a příjem signálu:

Na micro:bitu označeném *vysílač* sestaví scénář:



Obdobně na micro:bitu *přijímač*:



2. krok 15 minut

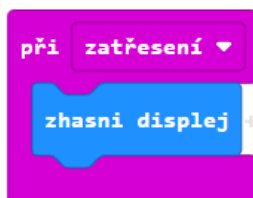
Nyní žáci vyzkouší **oba směry přenosu**.



2. úkol. Žáci přidají další scénáře, které zařídí, aby po stisknutí tlačítka A na druhém micro:bitu se přenesl signál na první micro:bit a zobrazilo se písmeno na displeji.

Řešením jsou úplně stejné scénáře jako v předchozím úkolu, scénáře se nahrají na oba micro:bity.

Žáci mohou přidat na každý monitor scénář, který reaguje na zatřesení a zhasne displej:



Žáci mohou měnit obrázky a texty, které se zobrazují po přijetí zprávy.

3. krok 15 minut

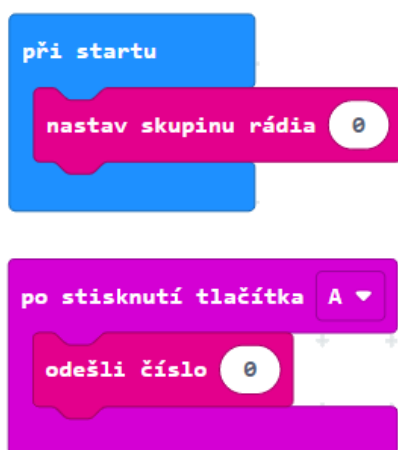


3. úkol. Žáci vytvoří program, který z micro:bitu udělá počítadlo „na dálku“. Bude to podobné počítadlo jako v kapitole 2. Po stisknutí tlačítka A *vysílač* vyšle číslo, *přijímač* po přijetí čísla přičte k proměnné jedničku a zobrazí **počet stisknutí** na displeji.

Žáci by měli být schopni program sestavit sami. S vytvořením proměnné jim můžete pomoci otázkou „Jak si bude micro:bit pamatovat, kolikrát se tlačítko stisklo“?

Ukázka možného řešení:

Vysílač



Přijímač



PRŮVODCE HODINOU V-2

Žáci se v této hodině naučí pomocí jednoho micro:bitu posílat a přijímat dvě rozdílné zprávy.

1. krok 10 minut

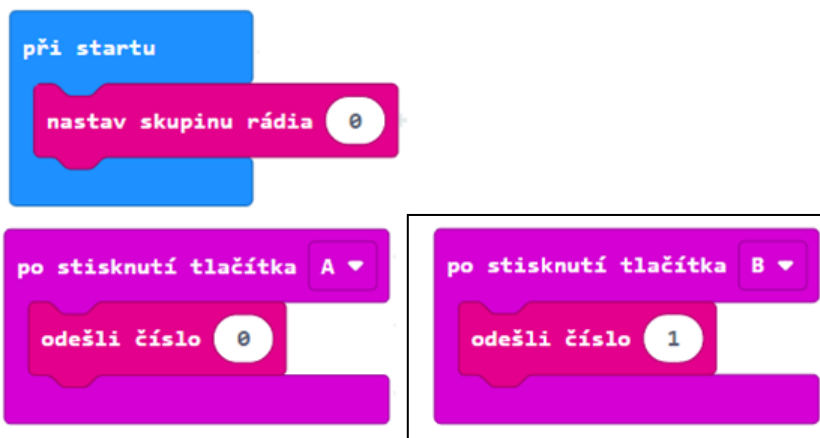
Nechte žáky, ať se rozdělí do skupin po dvou micro:bitech a ať se domluví, který z micro:bitů bude *vysílač* (bude vysílat signál) a který *přijímač* (bude přijímat signál). Nechte je, ať si spolu domluví skupinu rádia (vysílací kanál). Každá skupina musí mít odlišnou skupinu rádia, aby se vzájemně nerušily.

Posílání více zpráv. Žáci nejprve sestaví program z minulé hodiny, v němž po stisku tlačítka A se pošle zpráva a na *přijímači* se zobrazí písmeno „A“.



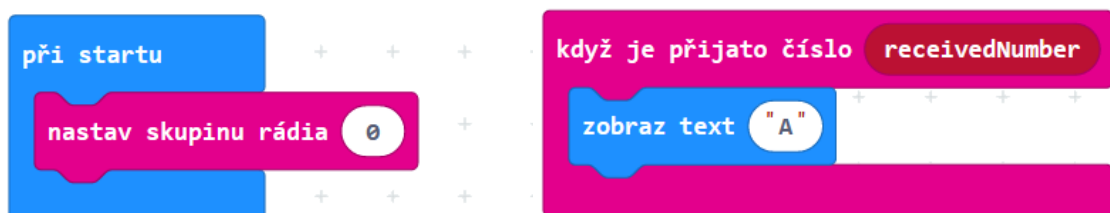
4. úkol. Zadejte žákům, aby do vysílače přidali scénář, tak aby micro:bit vysílal při stisknutí tlačítek A, B **dvě různé zprávy**.

Tento obrázek žákům neukazujte, na řešení mají přijít sami (v rámečku je přidán scénář):



Vysílač:

Přijímač po přijetí signálu bude mít za úkol zobrazit písmeno „A“. Ukázka řešení:



Přijímač:

Po sestavení se žáků zeptejte, jak jim přenos funguje a jestli poznají na displeji, které tlačítko se stisklo.

Pokud žáci vytvoří řešení jako na obrázku, zjistí, že *přijímač nerozezná*, jestli *vysílač* stiskl tlačítko A nebo B. Řešení tohoto problému budeme hledat v příštím kroku.

2. krok 10 minut

Zobrazení posílaného signálu. Pokračujeme z předchozího kroku.



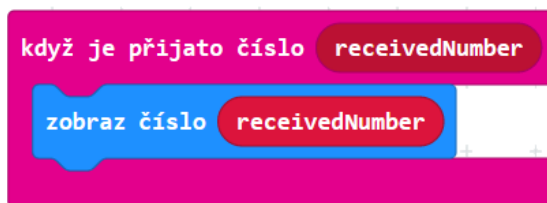
5. úkol. Nejdříve rozeberte se žáky tento scénář:



Diskutujte se žáky, co znamená proměnná **receivedNumber** (omlouváme se za takový název, ale v současné verzi Makecode jej nelze přeložit do češtiny). Žáci mohou mít nápad, že v proměnné je hodnota čísla, přijatého rádiem. Žákům správnou odpověď nesdělujeme – pokud na ni nepřijdou, nevadí, v dalším kroku to sami odhalí.

Pro kontrolu správnosti odpovědi (nebo pokud žáci na žádnou odpověď nepřišli) nechejte žáky **upravit scénář přijímače** tak, aby zobrazoval na displeji hodnotu proměnné **receivedNumber**.

Ukázka řešení:



Po nahrání upraveného programu do *přijímače* bude *přijímač* ukazovat 0, pokud bylo stisknuto tlačítko A, a 1 po stisknutí tlačítka B na vysílači. Žáci odhalí, že blok zobraz číslo ukáže na displeji číslo, které vysílač vyslal.

Z toho by žáci měli usoudit, že **receivedNumber** je proměnná, která uchovává hodnota čísla, přijatého rádiem.

Tento poznatek žákům nesdělujte, měli by na něj přijít sami. Můžete jim pomoci nápovědnými otázkami (Co se zobrazilo, když se stisklo tlačítko A? A co po tlačítku B? Co tedy proměnná **receivedNumber** uchovává?)

3. krok 10 minut



6. úkol. Požádejte žáky, aby *vysílač* po stisku tlačítek A, B **vysílal rozmanitá čísla** (nejen pouze 0 a 1).

Po vyzkoušení požádejte celou třídu, aby si u všech micro:bitů **nastavili skupinu rádia na stejné číslo 0** a vyzkoušeli. Žáci by měli vidět, že se jim na displeji *přijímače* nyní objevují i jiná čísla, než mají na svém vysílači, a to i tehdy, když jejich *vysílač* právě žádný signál nevyslal.

Žáci vlastně zjistí, že mohou „odposlouchávat“, co si jiní posílají. Aktivita může přejít v diskusi o bezpečnosti, o sledování komunikace mezi počítači a nutnosti šifrování.

Můžete přitom pohovořit o bezpečnosti internetové komunikace a typu útoku „man in the middle“.

Po této aktivitě zase každá skupina nastaví své původní číslo skupiny rádia.

3. krok 15 minut



7. úkol. Úkolem žáků bude upravit kód pro *přijímač* tak, aby na displej **vypsal, které tlačítko bylo na vysílači stisknuto**. Tedy zobrazí „A“, pokud je stisknuto tlačítko A, a „B“ při stisku tlačítka B.

Možná řešení pro *přijímač*:

```

když je přijato číslo receivedNumber
  když receivedNumber = 0 tak
    zobraz text "A"
  jinak
    zobraz text "B"
  
```

1)

```

když je přijato číslo receivedNumber
  když receivedNumber = 0 tak
    zobraz text "A"
  když receivedNumber = 1 tak
    zobraz text "B"
  
```

2)

Diskutujte se žáky nad různými možnostmi řešení. Tato řešení jim promítněte.

Např. pokud bychom chtěli z vysílače poslat další odlišný signál (třeba při zatřesení nebo stisknutí A+B současně), v programu 1) vlevo by se tento třetí signál zobrazoval jako stisknutí tlačítka B.

V programu 2) vpravo by přijímač na třetí signál nijak nereagoval.

Žáci už by měli být schopni posoudit výhody a nevýhody toho kterého řešení.

Náměty na úkoly pro rychlé žáky:

Variantou je namísto písmen zobrazovat obrázky (viz řešení vpravo).

Žáci mohou následně posílat další signály jako reakce na další události.

Pokročilejší žáci mohou dotvořit programy tak, aby oba micro:bity byly vysílačem i přijímačem a dalo se **posílat signály oběma směry**.

```

když je přijato číslo receivedNumber
  když receivedNumber = 0 tak
    ukaž ikonu [obrázek]
  když receivedNumber = 1 tak
    ukaž ikonu [obrázek]
  
```

PRŮVODCE HODINOU V-3

Žáci v této hodině pomocí micro:bitu zakódují krátkou zprávu, pošlou ji spolužákovi přes micro:bit a ten ji zase rozkóduje.

Tato hodina se dá protáhnout na dvě vyučovací hodiny v případě, že se žáci zaberou do kódování zpráv.

1. krok 15 minut

Opět připomeneme rozdělení do skupin po dvou micro:bitech a volbě unikátního čísla skupiny rádia pro každou skupinu.

Vysílání Morseovou abecedou.

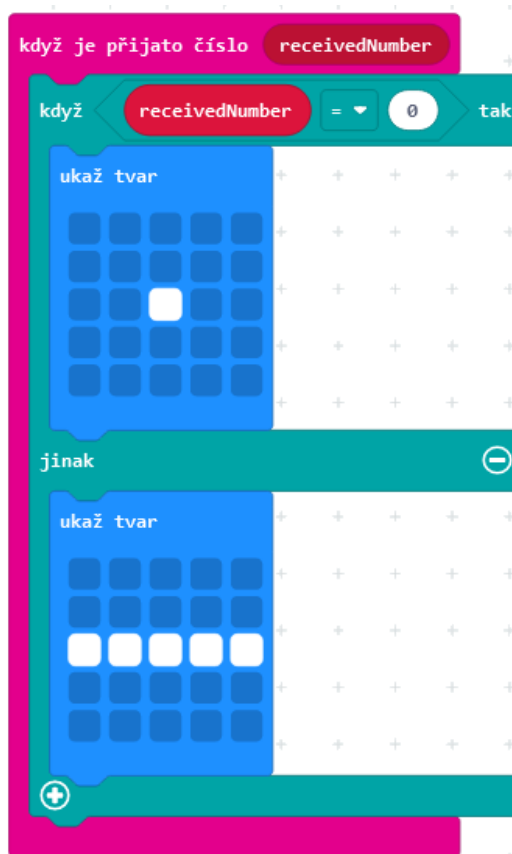
V úvodu se můžete zeptat žáků, jestli už někdy šifrovali zprávu pomocí Morseovy abecedy nebo jestli o ní alespoň slyšeli. Poté je nechejte najít na internetu tabulku s jejími kódy.

Nyní s žáky diskutujte, jak by se pomocí micro:bitu daly poslat zprávy zašifrovaná právě v Morseově abecedě. Žáci by měli navrhnout, že např. tečku lze posílat tlačítkem A a čárku tlačítkem B.



8. úkol. Zadejte žákům sestavit programy pro *vysílač* a pro *přijímač*. Žáci by měli přijít na to, že budou v podstatě stejné jako v minulé hodině, namísto písmen se v přijímači bude ukazovat tvar tečky nebo čárky.

Přijímač:



Vyzkoušejte. Žáci vysílačem pošlou kód vybraného písmena a na displeji přijímače přečtou kód a v tabulce poznají poslané písmeno.

2. krok 15 minut



9. úkol. Hledání a oprava chyby při přenosu. Zeptejte žáků na zkušenosti z předchozího úkolu, jestli posílání písmen Morseovou abecedou podle nich nemá nějaký problém. Žáci by mohli odhalit, že nelze rozeznat více poslaných teček za sebou (nebo čárek za sebou).

Pokud žáci chybu neodhalí, neříkejte správnou odpověď, ale rozdejte žákům, kteří vysílají, papírky s kódem písmena D (-..) nebo U (-.-) s tím, aby je poslali. Papírky rozdáváme proto, aby příjemci písmeno dopředu neznali. U *přijímače* žáci patrně zachytí písmeno N (-.) či A (-.).

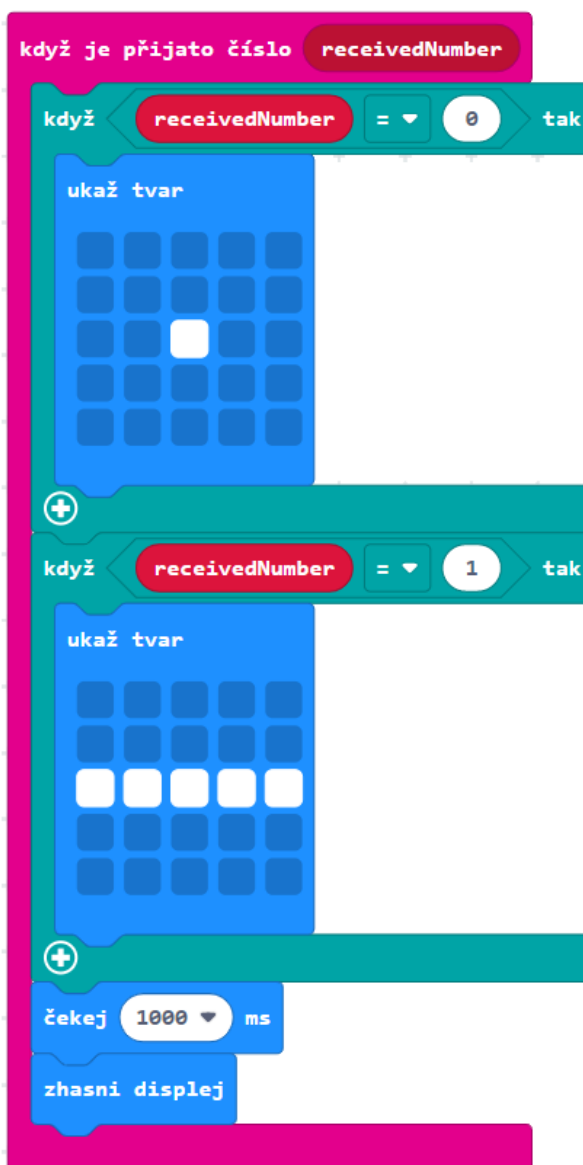
Diskutujte se žáky, jak odstranit tento problém. Dobrým návrhem je **nechat svítit displej jen chvíli**, třeba sekundu, a pak jej zhasnout.

Někdo z žáků může mít nápad rozsvítit na chvíli i displej *vysílače*, aby dělal mezi stisknutími tlačítek prodlevy a žáci u přijímače stihli znak přečíst. Pochvalte jej a doporučte toto řešení.

Příklad řešení pro *vysílač*:



Příklad řešení pro *přijímač*:



3. krok 15 minut



10. úkol - posíláme slovo. Zde budeme zkoumat, jestli lze naším programem bezpečně posílat celá slova, případně jak program opět vylepšit.

Zadejte žákům poslat jednoslovnou zprávu zašifrovanou pomocí Morseovy abecedy bez toho, aby žáci vysílající zprávu mluvili se žáky, kteří ji přijímají.

Opět se zeptejte se žáků, jestli neměli nějaké potíže. Měli by odhalit problém, jak rozeznat, že jedno písmeno končí a druhé začíná. Pokud žáci chybu neodhalí, nechejte je poslat slovo TIE (kravata). Patrně doručí zprávu TS (E s jednou tečkou a I se dvěma tečkami se spojí v S se třemi tečkami) nebo zprávu B (čárka a tři tečky).

Nechejte žáky, aby problém odstranili sami. Měli by vyloučit řešení pomocí časové prodlevy mezi slovy, protože to je nespolehlivý způsob. Patrně se dohodnou poslat mezeru mezi písmeny jako speciální znak, třeba po stisknutí A+B najednou (ostatně Morseovka to tak dělá, používá znak |). Simulátor micro:bitu v Makecode dokáže simulovat stisk obou tlačítek najednou (tlačítko A+B) i zatřesení (tlačítko SHAKE).

Podobným problémem je posílání mezery mezi slovy nebo konce celé zprávy. Řešením je opět poslat další speciální znaky.

Řešení zde uvedené je jedno z možných, posílá mezeru mezi písmeny a konec zprávy.

Vysílač:

Jsou přidány dvě události (číslo 2 znamená konec písmene, číslo 3 konec zprávy)

```

    po stisknutí tlačítka A+B
      odešli číslo 2
      ukaž ikonu [Morse]
      čekej 1000 ms
      zhasni displej
  
```

```

    při zatřesení
      odešli číslo 3
      ukaž ikonu [Morse]
      čekej 1000 ms
      zhasni displej
  
```

Přijímač:

```

    když je přijato číslo receivedNumber
      když receivedNumber = 0 tak
        ukaž tvar [Morse]
      jinak když receivedNumber = 1 tak
        zobraz text "-"
      jinak když receivedNumber = 2 tak
        zobraz text "/"
      jinak když receivedNumber = 3 tak
        ukaž ikonu [Morse]
      +
      čekej 1000 ms
      zhasni displej
  
```

PRŮVODCE HODINOU V-4, 5



11. úkol. Vytvoření systému pro anketu. Dvouhodinová aktivita pro pokročilé.

Žáci se v této hodině naučí posílat zprávy více micro:bitům zároveň. Po rozdání micro:bitů žákům představte cíl dnešní hodiny, vytvořit anketu plně ovládanou pomocí micro:bitů.

V žákovských listech jsou jednotlivé kroky řešení uvedeny otázkou, která se v daném kroku má řešit.

Jde o komplexní úkol pro pokročilé žáky. Budeme jej řešit v několika krocích, v nichž se nejprve vymyslí návrh, jak má být celý systém realizován, a v dalších krocích se sestaví vždy část programu. Budeme postupovat chronologicky, jak bude hlasování probíhat, takže budeme přecházet mezi programováním pro dva různé micro:bity.

Je možné třídu rozdělit na dvojice micro:bitů. Druhou možností je řešit úkol společně, před projekcí (sestavovat program před ostatními může vždy jeden žák, který se prostřídá, ostatní budou radit).

Bude potřeba, aby učitel jednotlivé kroky dobře řídil. Pro žáky je obtížné procházet celým úkolem samostatně.

1. krok 15 minut

Nejprve je třeba stanovit, **co vlastně má systém pro řízení ankety dělat**. O tom je možné s žáky diskutovat, nebo zadání rovnou žákům říci.

Zadání může znít: budeme mít jeden centrální micro:bit a několik hlasovacích micro:bitů. Centrální micro:bit bude shromažďovat údaje (hlasy), poslané z hlasovacích micro:bitů. Hlasovací micro:bity budou mít všechny stejný program. Budou hlasovat tak, že buď se stiskne tlačítko A (jsem pro návrh) nebo B (jsem proti návrhu). Centrální micro:bit hlasy zaznamená, sečte a na požádání zobrazí.

Na počátku je třeba vyřešit následující problémy:

1. Jak hlasovací micro:bit pozná, že už se smí hlasovat?
2. Jak zařídit, aby každý micro:bit mohl hlasovat pouze jednou?
3. Jaké různé signály se budou posílat?

Na první otázku mohou žáci odpovědět, že by centrální micro:bit měl vyslat signál, že je hlasování zahájeno.

Na druhou otázku mohou najít řešení, že posláním signálu o startu hlasování se každý hlasovací micro:bit „nabije“ jedním hlasem a hlasováním se tento hlas vyčerpá, vynuluje.

U třetí otázky je třeba se dohodnout, co bude znamenat který signál, poslané číslo. Protože všechny micro:bity budou přijímat i vysílat, nesmí se poslat z centrálního i z hlasovacího micro:bitu stejné číslo. Jinak by mohl micro:bit reagovat na signál, který vyslal on sám.

Návrh (který použijeme v našem řešení):

0 – hlasuji A (pro)

1 – hlasuji B (proti)

2 – start hlasování

Konec hlasování není třeba nastavovat, protože každý hlasovací micro:bit bude mít jeden hlas, a tak hlasování skončí automaticky, až všichni odhlasují.

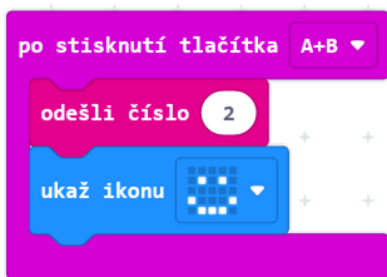
2. krok 10 minut

Start hlasování na centrálním micro:bitu.

Po stisknutí nějakého tlačítka centrální micro:bit vyšle signál, že je hlasování zahájeno. Může to zobrazit i na displeji (např. smajlíkem).

Protože tlačítka A a B si budeme šetřit na zobrazení počtu hlasů pro a proti na displeji, můžeme spustit start hlasování stisknutím A + B současně (nebo jinou událostí).

Ukázka řešení dílčího kroku – centrální micro:bit:



3. krok 10 minut

Start hlasování na hlasovacím micro:bitu.

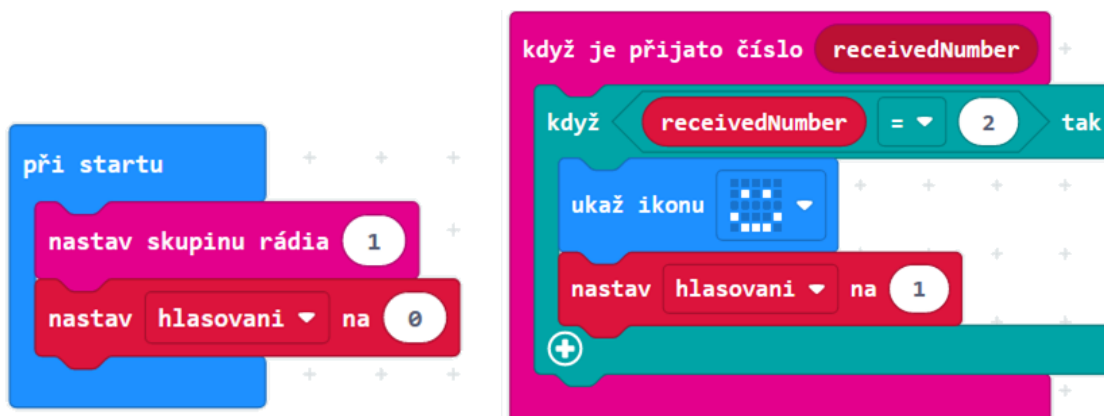
Pokud programujeme společně před žáky, musíme nyní přepnout na „jiný micro:bit“, např. na jinou webovou stránku s Makecode. Budeme tak mít dvě stránky, na jedné budeme vytvářet program pro centrální micro:bit, na druhé pro hlasovací.

Hlasovací micro:bit si potřebuje pamatovat, jestli smí nebo nesmí hlasovat. K tomu využijeme proměnnou **hlasování**. Hodnota 1 znamená, že je hlasování dovoleno, hodnota 0, že je zakázáno.

Po spuštění se hlasování nastaví na 0 a po přijetí signálu, že je hlasování zahájeno, nastaví hlasování na 1. Po odhlasování opět nastaví hodnotu na 0. Protože centrální micro:bit odstartuje hlasování jen jednou, hlasovací micro:bit jen jednou nastaví hodnotu na 1 a bude tak mít jen 1 hlas.

Po přijetí čísla rádiem ovšem musí hlasovací micro:bit zkontrolovat, jestli přišel správný signál (číslo 2 – start hlasování). Micro:bit totiž bude přijímat i signál od ostatních micro:bitů, které budou hlasovat, a mohl by tak reagovat na nevhodnou událost. Zobrazením smajlíku na displeji dává najevo, že hlasování začalo.

Takto může vypadat program pro start hlasování na hlasovacím micro:bitu:



4. krok 10 minut

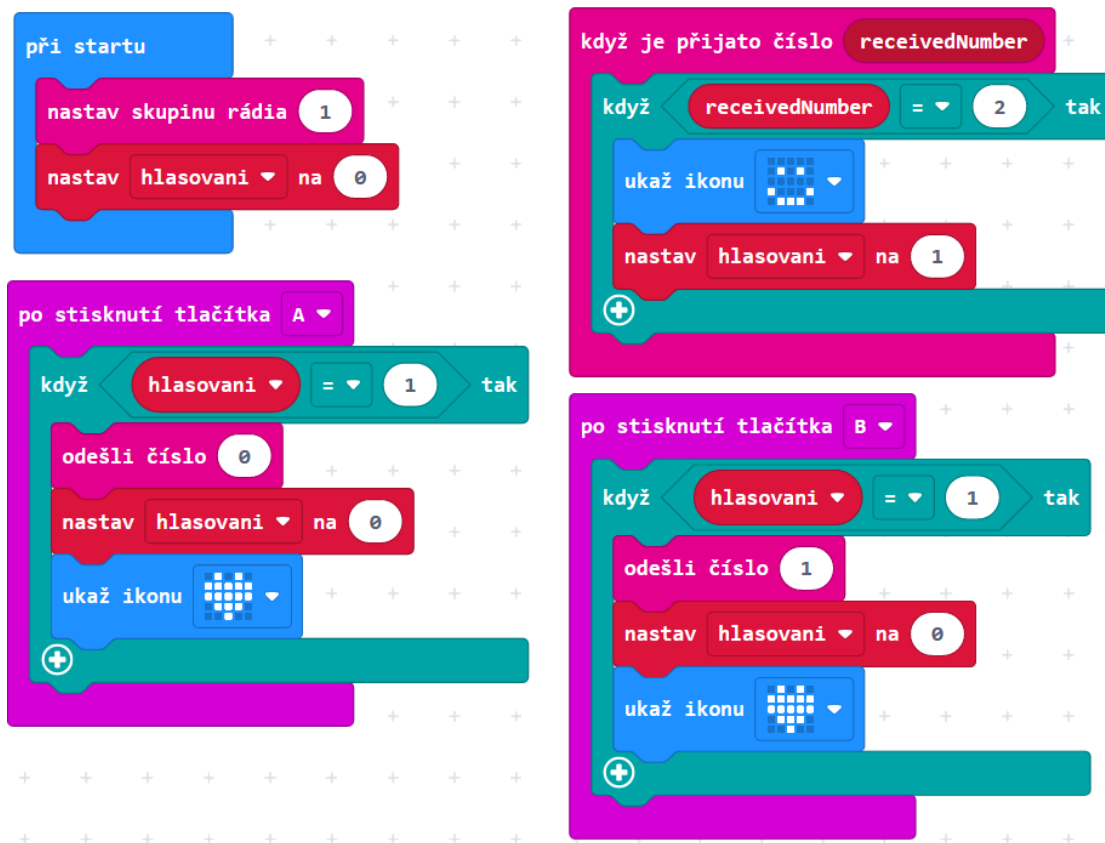
Hlasování na hlasovacím micro:bitu.

Po stisknutí tlačítka A, B je nejprve nutno zkontrolovat, jestli se hlasovat smí (tlačítka lze stisknout kdykoliv). Jestliže je hlasování povoleno (hodnota proměnné **hlasování** = 1), odešle zprávu, které tlačítko se stisklo, a musí nastavit hodnotu **hlasování** na 0, aby již hlasovat znovu nemohl.

Po hlasování můžeme ikonou (v našem případě srdíčkem) ukázat, že bylo odhlasováno.

Tyto poznatky o tom, jak má hlasování fungovat, mohou žáci odhalit sami nebo při diskusi. Chceme-li je naučit programovat, pak vezme, že když žákům poznatek řekneme (vysvětlíme), vlastně na něj nepřišli sami. I Vás více potěší, když žáci jednotlivé dílčí úkoly vyřeší sami. Proto „vysvětlení“ nebo nápovědu volíme až v krajním případě.

Ukázka možného programu pro hlasovací micro:bit po doplnění scénářů pro hlasování:



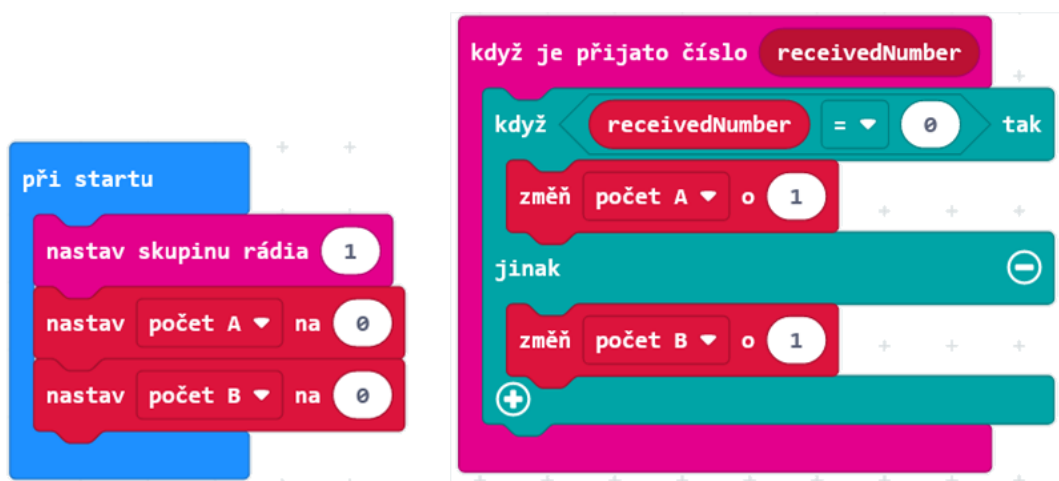
5. krok 15 minut

Počítání hlasů na centrálním micro:bitu.

Centrální micro:bit si musí pamatovat, kolik hlasů hlasovalo pro (A) a kolik proti (B). K tomu budeme potřebovat dvě proměnné, **počet A** a **počet B**. Tento micro:bit musí přijímat zprávy od ostatních a rozlišit, jestli hlasují pro (hodnota **hlasování** = 0) nebo proti (**hlasování** = 1). Potom zvětší hodnotu správné proměnné o 1.

Na startu ovšem musí vynulovat hodnoty proměnných **počet A**, **počet B** a také nastavit skupinu rádia.

Ukázka možného řešení dílčího kroku:

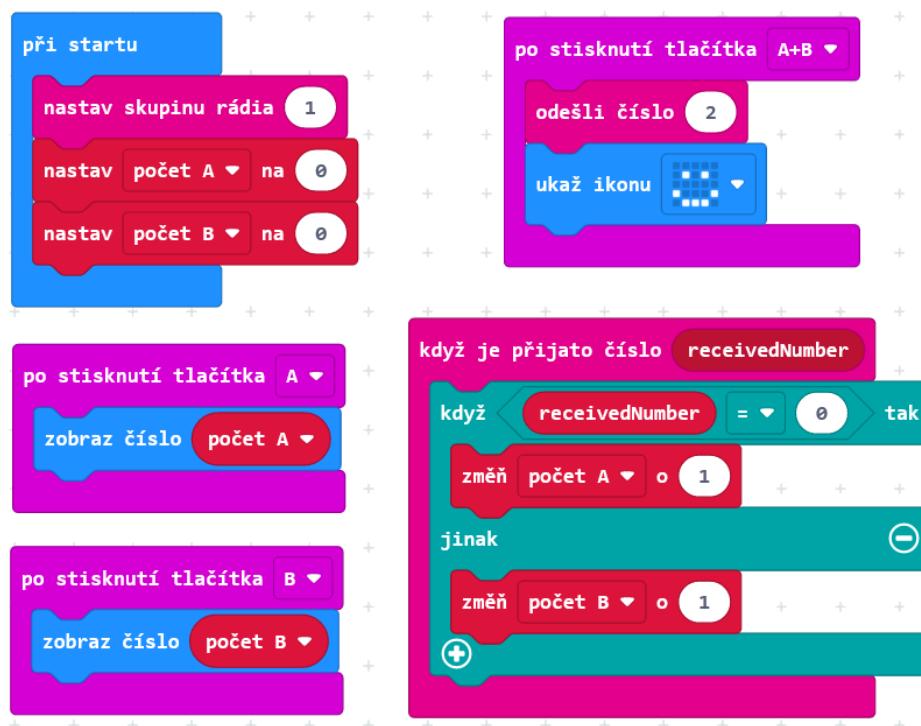


6. krok 10 minut

Zobrazení hlasů na centrálním micro:bitu.

Zobrazení bude jednoduché: po stisknutí tlačítka A se na displeji zobrazí hodnota proměnné **počet A**. Obdobně po stisknutí B se zobrazí hodnota **počet B**.

Ukázka možného programu pro centrální micro:bit po doplnění všech potřebných scénářů - vpravo:



7. krok 20 minut

Vyzkoušení systému pro hlasování.

Přehrajte se žáky, jak anketa bude probíhat. Do všech micro:bitů kromě jednoho (centrálního) nahrajeme program pro hlasování a všude nastavíme stejnou skupinu rádia. Pak centrální micro:bit odstartuje hlasování a všechny ostatní micro:bity hlasují. Na konci centrální micro:bit zobrazí počty hlasů pro a proti.

Hlasování několikrát opakujte na různá zajímavá témata. Vždy předejte centrální micro:bit k ovládní jinému žákovi.

Náměty na témata:

- Máš rád sladká jídla? (ano/ne)
- Věříš, že existují mimozemšťané?
- Kouříš? (hlasování je anonymní; bude zajímavé sledovat, jak to žáci pojmu)
- Byl bys pro, aby se zrušilo zkoušení ve škole?
- Vnímáš změny klimatu jako důležitou věc a problém?
- Zrušil bys mince a papírové peníze?

Poznámka: Je možné, že pokud dva žáci odešlou signál současně, může dojít ke ztrátě dat (přijímač přečte jen jeden). To je problém, který bychom na této úrovni těžko vyřešili, ale pokud nastane, můžeme jej okomentovat, že v reálné situaci (např. online volby prezidenta) by toto muselo být také vyřešeno.

6 PŘIPOJOVÁNÍ PERIFÉRIÍ

Tato kapitola se věnuje připojení periférií, dalších součástek, ze kterých budeme získávat data nebo jejichž chování budeme pomocí micro:bitu ovládat. Konkrétně budeme ovládat tříbarevnou diodu a získávat data z externího teplotního čidla.

Kapitola je koncipována jako volitelná. Ponecháváme zcela na vyučujícím, zda se ji rozhodne zařadit.

Zapojení externích prvků znamená další finanční náklady v řádu do 100 Kč na žáka. Umožní však atraktivnější aktivity – žáci uvidí, že k centrální desce lze připojovat další vstupní a výstupní zařízení a čidla a informační systém tak „stavebnicově“ doplňovat.

Úlohy lze zapojovat i pomocí nepájivého pole a přiblížit se tak více elektrotechnické praxi. Necháme na učiteli, zda touto cestou půjde.

Co se naučíte

- Připojit a ovládat tříbarevnou diodu
- Připojit teplotní čidlo a zjistit pomocí něj teplotu
- Pomocí dvou micro:bitů sestavit jednoduchý bezdrátový teploměr se vzdálenou sondou

Co budete potřebovat

- PC s internetovým připojením
- Propojovací USB kabel micro USB koncovkou
- Micro:bit
- Čtyři vodiče nejlépe s krokodýlky na obou koncích
- Tříbarevnou diodu se společnou katodou (záporným pólem) – obr. vlevo
- Teplotní čidlo pro napětí 3 V, např. TMP-36 – obr. vpravo

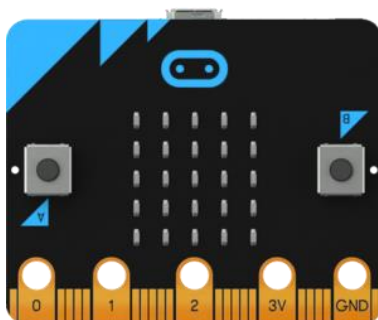


Časová náročnost

- 4 vyučovací hodiny po 45 minutách (včetně závěrečného projektu 5 hodin).

PRŮVODCE HODINOU VI-1

Žáci připojí k micro:bitu externí součástku, tříbarevnou RGB LED diodou. Vyzkouší si zápis logických a číselných hodnot.



Úvodní teorie pro učitele

Micro:bit je s periferiemi spojen pomocí tzv. pinů dole na desce. Pět nejvíce využívaných pinů lze zapojit pomocí krokosvorek nebo jimi prostrčit banánky s kabely. Jsou označeny čísly 0, 1, 2, dále napájení 3V a zem GND.

Micro:bit po zapojení dokáže číst z připojených pinů hodnoty (logické 0, 1 nebo číselné od 0 do 1023) nebo na ně tyto hodnoty odesílat (zapisovat). Periferie si s těmito zapsanými hodnotami nějak poradí. Jejich význam může být u každé periferie jiný; obecně platí, že logická 0 je vypnuto, 1 zapnuto; číselné hodnoty pak znamenají různé množství (např. větší číslo může znamenat více světla, větší teplotu, rychlejší otáčky apod).

Čtení a zápis hodnot do pinů se v Makecode provádí pomocí bloků v sekci **Piny (Rozšířené)**. Pozor, na rozdíl od některých programovacích jazyků, např. Scratch, Makecode má pro čtení a zápis logické hodnoty jiné příkazy než pro čísla.

přečtená logická hodnota z pinu P0 ▾

přečtené číslo z pinu P0 ▾

zapiš do pinu P0 ▾ logickou hodnotu 0

zapiš do pinu P0 ▾ číslo 1023

Tato úvodní teorie **nemá** sloužit k tomu, aby s ní učitel žáky seznámil, aby ji na úvod odvykládal. Pro žáka není důležitá, cennější je, když na tyto poznatky bude vlastní činností přicházet a diskusí s ostatními je krystalizovat.

1. krok 20 minut

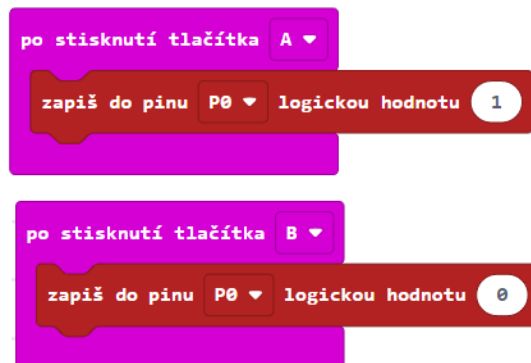


1. úkol. Žáci k micro:bitu připojí tříbarevnou diodu podle zadání:

Zapojte dva z pinů na tříbarevné diodě. Ten nejdlejší připojte k pinu GND na micro:bitu a libovolný další k pinu 0. Ostatní nepřipojujte.

6 PŘIPOJOVÁNÍ PERIFERÍÍ

Nechejte žáky **vyzkoušet** scénáře:

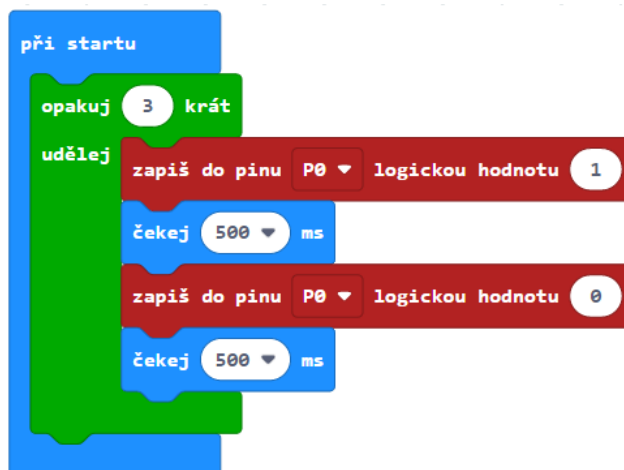


Žáci by měli říci, že tlačítko A rozsvítí diodu a tlačítko B ji zhasne.

Nejdelší pin na diodě nechejte, jak je, a požádejte žáky, aby u druhého pinu krokosvorku **přepojili k jinému pinu na diodě**. Žáci by měli odhalit, že dioda bude nyní při nezměněném programu svítit světlem jiné barvy. Tedy budou vědět, že program sám o sobě nerozsvěcuje diodu nějakou barvou – záleží na zapojení.

Poté žákům zadejte úkol, aby sestavili scénář, v němž dioda **několikrát zabliká**.

Takto může vypadat správné řešení:



2. krok 15 minut

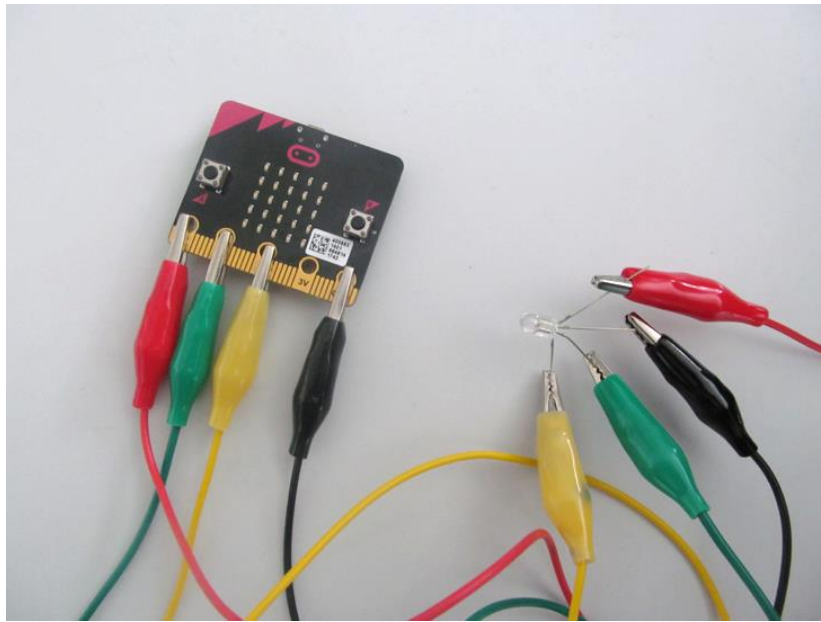
V tomto kroku připojíme najednou všechny tři barevné kanály na diodě. Budeme tedy potřebovat tři piny na micro:bitu a dávat pozor, na který pin se co posílá.



2. úkol. Žáci zapojí tříbarevnou diodu k micro:bitu podle obrázku. Zem (GND) zapojí na nejdelší pin tříbarevné diody. Ostatní zapojení je doporučeno: pin 0 k samostatnému pinu diody (červená), který je na jedné straně od nejdelšího pinu diody. Oba piny z druhé strany diody zapojíme tak, že pin blíže ke středu (zelená) připojíte k pinu 1 a poslední pin (modrá) k pinu 2. Viz obrázek.

Zapojování není snadné, na druhou stranu, když žáci něco spletou a barvy pomíchají, nic strašného se neděje. Musí ovšem nechat GND vést na nejdelší pin, aby dioda fungovala.

6 PŘIPOJOVÁNÍ PERIFÉRIÍ



Nechteje sestavit žáky scénář, který nejprve rozsvítí diodu na pin 0 a po jedné sekundě ji zhasne. Poté rozsvítí diodu na pin 1 (a opět ji po sekundě zhasne) a nakonec rozsvítí diodu na pin 2 a opět ji zhasne.

Ukázka řešení této úlohy:

```
při startu
zapiš do pinu P0 logickou hodnotu 1
čekej 1000 ms
zapiš do pinu P0 logickou hodnotu 0
zapiš do pinu P1 logickou hodnotu 1
čekej 1000 ms
zapiš do pinu P1 logickou hodnotu 0
zapiš do pinu P2 logickou hodnotu 1
čekej 1000 ms
zapiš do pinu P2 logickou hodnotu 0
```

Pokud je vše zapojeno v pořádku, měly by se postupně rozsvítit vždy na vteřinu postupně červená, zelená a modrá.

Pokud je pořadí barev jiné, nevadí – do správného pořadí je převedeme změnou zapojení pinů na diodě ke krokosvorkám.

3. krok 20 minut

V minulých úkolech jsme používali logické hodnoty zapnuto – vypnuto; dioda svítila naplno nebo zhasla. V této aktivitě využijeme číselných hodnot, které umožní postupné rozsvěcování diody.

Zapojení ponecháme stejné.



3. úkol. Necháme žáky vyzkoušet tento scénář:

```

při startu
  pro pořadí od 0 do 1023
    udělej
      zapiš do pinu P0 číslo pořadí
      čekej 2 ms
  
```

Pozor, je důležité zkontrolovat, aby žáci nepoužili blok pro zápis logické hodnoty.

Žáci uvidí, že se dioda postupně rozsvítí jednou barvou (červeně). Změnou doby čekání můžeme nastavit rychlejší nebo pomalejší rozsvěcení.

Můžeme vyvolat diskusi, co znamená, že postupně posíláme větší a větší číselné hodnoty na pin 0. Žáci by měli vidět souvislost s rostoucím číslem a množstvím svítivosti diody.



4. úkol. Dalším úkolem je doplnit scénář, aby dioda po rozsvícení opět postupně vyhasínala.

Ukázka možného řešení:

```

při startu
  pro pořadí od 0 do 1023
    udělej
      zapiš do pinu P0 číslo pořadí
      čekej 2 ms
  pro pořadí od 0 do 1023
    udělej
      zapiš do pinu P0 číslo 1023 - pořadí
      čekej 2 ms
  
```

6 PŘIPOJOVÁNÍ PERIFÉRIÍ

Žáci mohou s úlohou dále experimentovat: např. mohou změnit pin, na který posílají číselné hodnoty.

Mohou také na začátku scénáře rozsvítit ještě jiný pin a pozorovat pak barevné přechody.

Tato aktivita poskytuje spoustu zábavy: např. postupně rozsvěcet všechny tři barvy (na pinech 0, 1, 2) až do úplně bílé, různé blikání nebo barevné přechody.

V příští hodině budete vyrábět magickou lampu. Je vhodné, aby se žáci na tuto výrobu připravili. Je třeba připravit vhodné stínítko, např. válec z papíru nebo vyrobený na 3D tiskárně, podle možností. Lze použít i menší lampion.

PRŮVODCE HODINOU VI-2



5. úkol. Žáci v této hodině sestojí pomocí třibarevné diody tzv. magické lampy, která náhodně střídá tři barvy. Téma je na celou hodinu, proto jej dále nedělíme.

Je efektní připravit si předem vhodné stínítko, např. válec z papíru nebo vyrobený na 3D tiskárně, podle možností. Lze použít i menší lampion.

1. krok - celá hodina

Ponechte zapojení diody jako v minulé hodině a nechte žáky vytvořit následující program:

Úloha je komplexnější, doporučujeme nejprve se žáky projít a promyslet, jak takový program má fungovat.

Učitel může diskusi vést těmito otázkami k žákům a podnítit diskusi, jak dílčí problémy řešit.

- Víme, že lampa má vždy náhodně vybrat nějakou barvu a tu rozsvítit. Žáci mohou navrhnout uchovat tuto náhodnou hodnotu v nějaké proměnné, např. **barva**.

- Když program vybírá novou barvu k rozsvícení, mohla by se vybrat tatáž barva, která již svítí. Je tedy třeba zařídit, aby když program vybere stejnou barvu, jako je minulé, začal vybírat znovu. Žáci mohou navrhnout použít blok **dokud** – nová barva se bude vybírat tak dlouho, než bude odlišná od minulé. Minulou barvu si také budeme muset pamatovat v proměnné, např. **minulá**.

- Jak program vybranou barvu rozsvítí? Žáci by mohli přijít na to, že podle toho, jako barvu program našel, musí zapsat do pinu, který v zapojení odpovídá příslušné barvě, logickou hodnotu 1.

- Když program rozsvěcí jinou barvu, musí tu původní zhasnout. Jak ale vyřešit, na který pin poslat logickou hodnotu? Žáci mohou navrhnout, že stejně jako při rozsvícení barvy může program zjistit podle hodnoty minulá, na který pin hodnotu zapsat. Nebo do každé větve bloku `když` přidat čekání a zhasnutí daného pinu. Nebo mohou přijít na jednodušší řešení, prostě zhasnout všechny piny a pak rozsvítit ten nový.

- Mezi rozsvícení a zhasínání diody musíme dát čas na čekání. Jak to udělat, aby tento čas nebyl pořád stejný? Žáky napadne dát náhodnou dobu čekání – tu je třeba rozumně nastavit.

Nechte žáky vyrobit skutečnou lampu, například jenom jako váleček ze čtvrtky, kde jednotlivé piny prostrčíte čtvrtkou ven. Důležité je, aby se jednotlivé konektory navzájem nedotýkaly.

Odbočka pro učitele:

Všimněte si, jak kvalita programovacího jazyka usnadní nebo neusnadní programování. Na další straně v ukázkovém řešení je celá střední pasáž věnována přidělování pinu podle hodnoty proměnné **barva**. Je na to spotřebováno 6 řádků kódu. Kdyby Makecode umožňoval jako parametr pro pin

hodnotu proměnné, tedy **pokud by umožňoval** takovýto příkaz:



bylo by možné zapsat celou tuto pasáž do jednoho přehledného bloku. Museli bychom ovšem více hlídat, jaká hodnota je v proměnné barva, abychom nezpůsobili chybu (např. pro hodnotu 1023 bychom zapisovali na pin 1023, který nemáme).

Tato odbočka není určena žákům.

6 PŘIPOJOVÁNÍ PERIFERIÍ

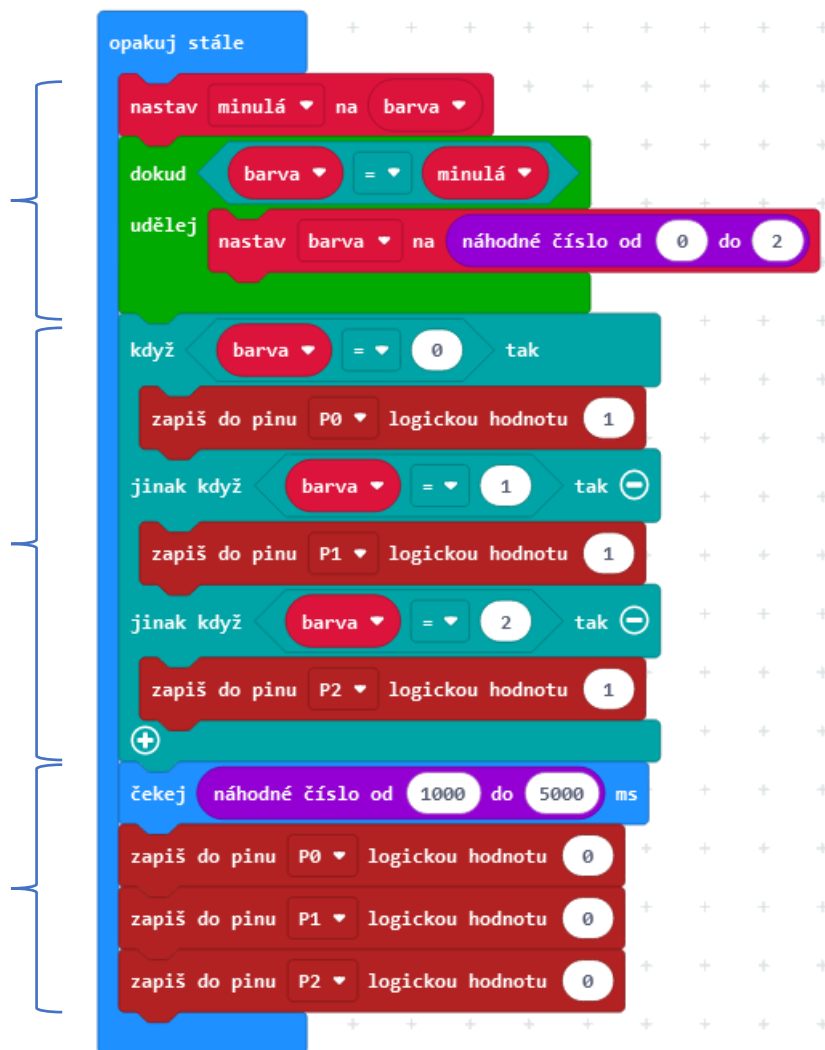
Ukázkové řešení:

Vysvětlivky:

Zde se vybírá náhodná barva (tak dlouho, až se bude lišit od minulé barvy). Předtím je třeba minulou barvu nastavit.

Zde se podle vybrané barvy vybírá pin, který se „rozsvítí“ (na který se pošle logická hodnota 1).

Po náhodné chvilce čekání se všechny piny pro jistotu „zhasnou“.



Rozšíření:

Rychlejší žáky vyzvěte, aby do programu přidali další barvy zapnutím dvou nebo více pinů najednou.

Žáci mohou vytvořit stroboskopickou lampu, která bude střídát barvy rychle nebo bude rozsvěcet a zhasínat.

PRŮVODCE HODINOU VI-3

Žáci připojí k micro:bitu externí součástku - teplotní čidlo. Vyzkouší si zpracovat číselný vstup z pinu.

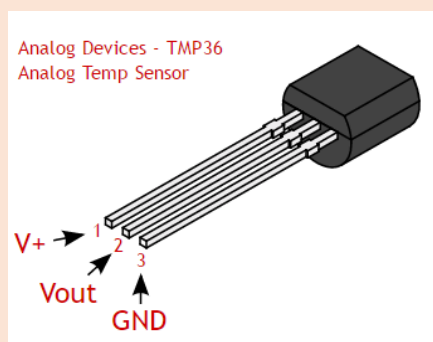


6. úkol. Žáci sestrojí digitální teploměr.

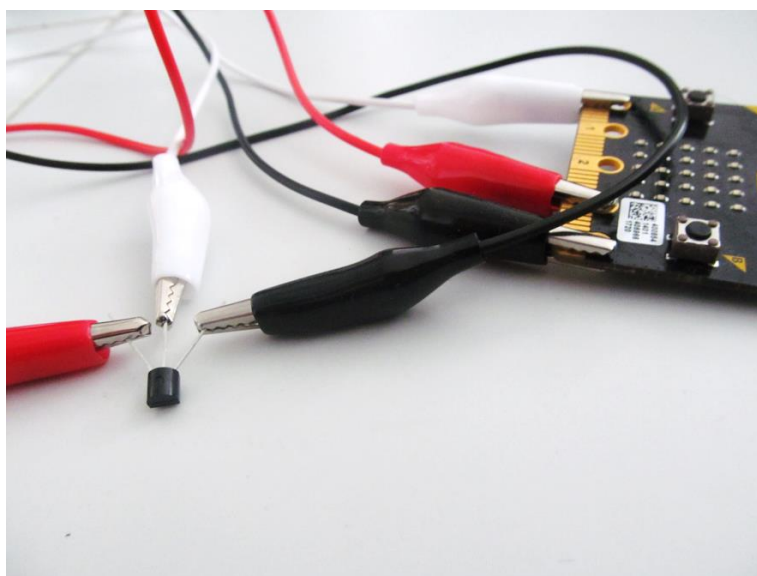
Pro tuto hodinu potřebujete nějaké levné teplotní čidlo, pracující s napětím 3 V. Předváděno je číslo TMP-36, ale je samozřejmě možné použít libovolné, které je k dispozici. K použitému čidlu potřebujete dokumentaci (zvanou datasheet), kterou buď dostanete spolu s čidlem, nebo si jí stáhnete z webu prodejce či odjinud z internetu.

1. krok 10 minut

První věc, kterou si musíte zjistit, je zapojení čidla. Například čidlo TMP-36 se zapojuje dle následujícího schématu:



Pin V+ je napájení, připojte jej na pin 3V na micro:bitu. GND (zem) připojte na pin GND. Vout je výstup, který zapojíte na libovolný pin, například na pin 0. Zapojení je ukázáno na následujícím obrázku. Všimněte si na fotografii, že plochá hrana čidla je dole. Dejte si pozor, abyste nespletli zapojení napájení a země, snadno byste mohli teplotní čidlo zničit. Nechte žáky číslo zapojit a před připojením micro:bitu ke zdroji jim zapojení zkontrolujte.



Zapojení teplotního čidla k pinům na micro:bitu.

2. krok 15 minut

Zde je pro učitele popsán princip funkce čidla.

Čidlo po připojení napájení a země začne měřit teplotu a výsledek sděluje velikostí napětí na výstupním pinu (V_{out}), jež nabývá hodnot od 0 do 1023 mV. Toto číslo ukazuje procento ze vstupního napětí, které je u micro:bitu 3.18 V, jak můžete ověřit Voltmetrem.

Proto je výpočet napětí:

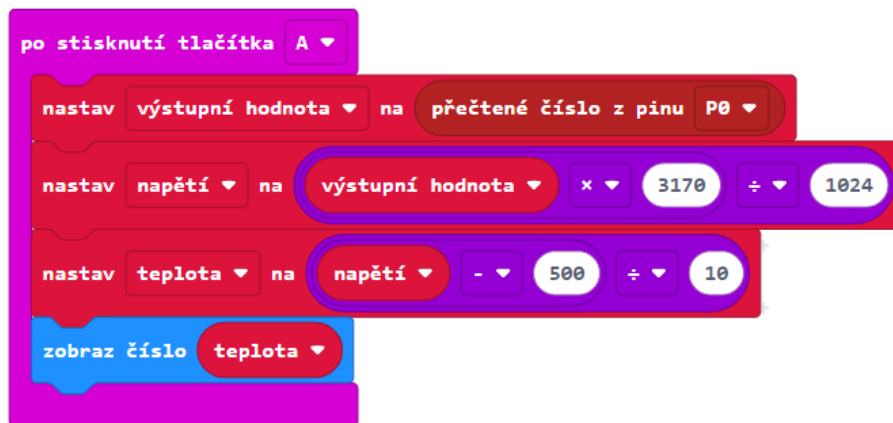
$$napeti = \frac{V_{out} \cdot 3.18}{1024}$$

Odtud pak již vypočtete teplotu (ve stupních Celsia):

$$teplota = \frac{napeti - 500}{10}$$

Tento vzorec je převzatý z datasheetu (manuálu) k čidlu TMP 36 a bude se pravděpodobně lišit, pokud máte jiné čidlo než popisované TMP-36. V takovém případě si naleznete potřebné vzorce v odpovídajícím datasheetu.

Žákům stačí vědět, že micro:bit teplotu ze vstupní hodnoty na pinu 0 vypočítá. Protože výpočetní vzorce jsou vzhledem ke znalostem žáků obtížné k porozumění, dovolte žákům **opsat si scénář**, který výpočet naměřené teploty z výstupního napětí provede.



Po každém stisknutí tlačítka A se naměřená teplota zobrazí na displeji.

Vysvětlete žákům, že po zapojení chvíli trvá, než se teplotní čidlo vyrovná s teplotou měřeného okolí, zejména pokud jej před použitím drželi delší dobu v ruce. První výsledky doporučujeme ignorovat. Teplota se postupně ustaluje na určité hodnotě.

Nechte žáky ověřit měřenou teplotu vzájemně, případně s jiným teploměrem. Pokud se výsledky významně liší, zkuste některý z následujících kroků:

- ověřte program, zejména výpočty (pořadí výpočtů ve vnořených blocích matematických operací) – v Makecode je dost těžko čitelné, který blok je uvnitř kterého.
- ověřte zapojení
- pokud se liší u všech žáků od spolehlivého teploměru pak ověřte, zda máte správný datasheet a používáte správný vzorec pro výpočet
- ověřte voltmetrem napětí

Rozšiřující úloha: Vytvořte scénář, který bude teplotu měřit automaticky po 10 sekundách a trvale ji zobrazovat.

Možné řešení:



3. krok 20 minut



7. úkol. Digitální teploměr se vzdálenou sondou. Využijí se zde zkušenosti z kapitoly 5.

Použijeme ve skupině dva micro:bity. Jeden bude teploměrem: bude mít ovládání a bude zobrazovat teplotu. Druhý bude sondou, bude připojen k teplotnímu čidlu a bude odesílat rádiem naměřené hodnoty do micro:bitu – teploměru.

Sondu pak budeme moci dát za okno a uvnitř místnosti sledovat venkovní teplotu.

Využijeme zapojení z předchozích kroků, pouze přidáme další micro:bit.

Každá skupina dvou micro:bitů musí mít unikátní skupinu rádia, aby v přenosech nebyl zmatek.

Nejprve se žáky vymyslete postup. Žáci by měli mít nápady, jak takový teploměr zkonstruovat a jak jej naprogramovat.

Očekávaný postup je:

- micro:bit – teploměr odešle rádiem micro:bitu – sondě zprávu, která znamená pokyn k měření
- micro:bit – sonda změří a vypočítá teplotu a pošle ji rádiem micro:bitu – teploměru
- micro:bit – teploměr zobrazí tuto hodnotu na displeji.

Způsob spouštění teploměru a zobrazení teploty si žáci mohou zvolit (např. teplota se změří po stisknutí tlačítka A, po zatřesení nebo se bude měřit průběžně po nějakém intervalu).

Úloha se dá rozšířit na vytvoření tzv. minimálního teploměru (v proměnné si pamatuje nejnižší dosud naměřenou teplotu, kterou zobrazí třeba po stisknutí tlačítka), podobně lze ukládat maximální teplotu.

Možné řešení:

micro:bit – teploměr



micro:bit - sonda



4. krok 30 minut



8. úkol. Signalizátor teploty na dálnici. Potřebujeme dva micro:bity, teplotní čidlo a trojbarevnou diodu.

Žáci sestaví základovou stanici a teplotní sondu, která bude měřit teplotu pomocí externího čidla podobně jako v předchozím úkolu. Pomocí rádia bude výsledky v určitém intervalu (např. 10 sekund) předávat základové stanici. Základová stanice bude zobrazovat naposledy obdrženu teplotu a podle naměřené teploty diodou signalizovat stupeň zahřátí vozovky (nízká – modrou barvou, vysoká – červenou barvou, akorát – zelenou barvou).

Hranice pro změnu barvy signalizace si nastavíme podle místních podmínek ve třídě.

Pro reálnou situaci na dálnici mohou být hranice méně než 3 °C – modrá, nebezpečí náledí, 3 °C až 20 °C – zelená, vyšší než 20°C – červená, riziko vyššího opotřebení pneumatik).

Pro potřeby ověření ve třídě nastavíme teploty tak, aby se dalo dosáhnout všech tří stupňů signalizace (umístěním na sluníčku, za oknem apod.) a přenášením micro:bitu. Pak mohou být hraniční teploty např. 18°C, 23°C – je třeba experimentovat.

Teplotní čidlo je možné pro vyzkoušení ohřívat a chladit, ale ne přímo ve vodě – hrozí zkrat.

Ukázka možného řešení:

```

když je přijato číslo receivedNumber
  zobraz číslo receivedNumber
  když receivedNumber > 20 tak
    zapiš do pinu P0 logickou hodnotu 1
  jinak když receivedNumber > 3 tak
    zapiš do pinu P1 logickou hodnotu 1
  jinak
    zapiš do pinu P2 logickou hodnotu 1
  
```

Řešení je stejné jako u předchozího úkolu, pouze je rozšířen scénář pro teploměr (základovou stanici) po přijetí naměřené teploty od sondy.

ZÁVĚREČNÝ PROJEKT

Učebnice je zakončena závěrečným projektem komplexnější povahy, při nichž žakovské týmy vytvoří věrnější napodobeniny reálných digitálních zařízení s čidly a výstupy.

Na projekt je třeba počítat minimálně s jednou vyučovací hodinou.



9. úkol. Zabezpečovací zařízení.

Žáci ze dvou micro:bitů sestaví zabezpečovací zařízení, které bude hlásit mimořádnou situaci. Jeden micro:bit bude fungovat jako čidlo a druhý jako základová stanice. Podle volby žáka může kontrolovat změnu polohy micro:bitu, změnu intenzity světla, teploty, magnetického pole apod.

Nabízí se tyto možnosti (nebo jejich kombinace), na co bude čidlo reagovat:

- Teplota se zvedla (klesla) pod určitou mez.
- Reaguje na změnu polohy. Např. micro:bit je umístěn na dveřích, které se otevřou, nebo v hlídaném předmětu, který je zvednut (otočen). Zde se nabízí využití kompasu (změna směru) nebo události při naklonění micro:bitu.
- Reaguje na významnou změnu intenzity světla (rozsvíceno, zhasnuto) pomocí bloku `intenzita světla`.
- Reaguje na významnou změnu magnetického pole. Např. je umístěn na dveřích, které se vzdálí od kovového rámu. Nebo se od něj vzdálí magnet umístěný v hlídaném předmětu. Žáci by museli prozkoumat blok `magnetická síla` v sekci *Vstupy*.
- Je stisknuto tlačítko – nouzový signál.

Ve všech těchto případech čidlo vyhlásí poplach (pošle signál základové stanici). Ta podle vybavení rozsvítí diody na displeji, zabliká trojbarevnou diodou nebo vyšle zvukový signál. Stiskem tlačítka na základové stanici je možné poplach odvolat.

Náročnějším rozšířením je požadavek, aby se čidlo pravidelně hlásilo, a pokud tak neučiní (např. při poruše), měl by poplach být rovněž vyhlášen.

Pokročilejší žáci mohou sestavit zabezpečovací zařízení, které bude kombinovat více prvků ochrany (např. hlídá teplotu, a ještě umožňuje spustit poplach stisknutím tlačítka).

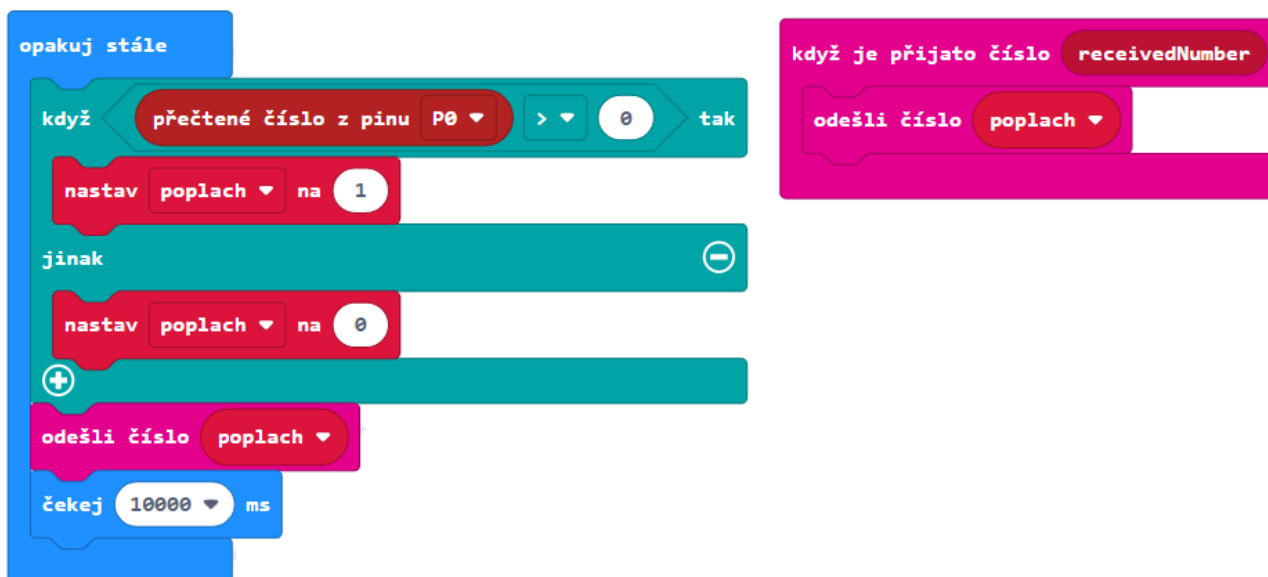
Neuvádíme zde vzorová řešení, protože pro mnoho variant projektu existuje mnoho variant řešení.

Ukázka možného řešení pro sondu je na další stránce. Sonda v pravidelných intervalech testuje vstup z pinu (z připojeného externího zařízení) a porovnává ji s nějakou prahovou hodnotou, která spouští poplach. Proměnná **poplach** nese informaci, jestli poplach je (1) nebo není (0). Tuto hodnotu sonda odesílá.

Pravý scénář ukazuje odpověď sondy na dotaz základové stanice, jestli sonda funguje. Odesílá hodnotu proměnné **poplach**.

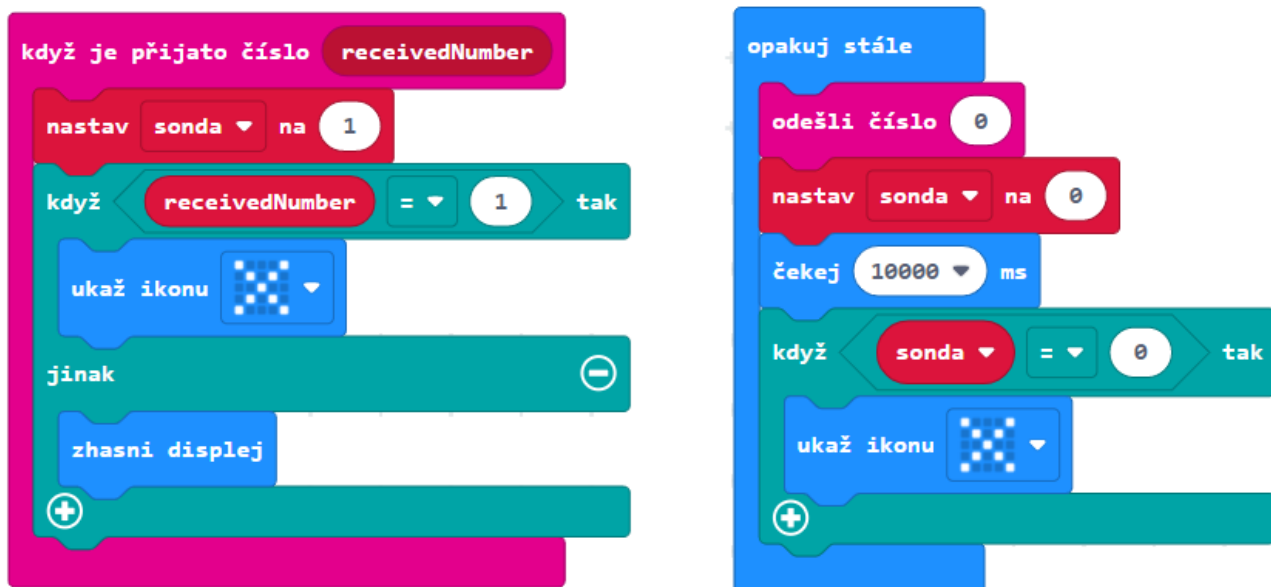
6 PŘIPOJOVÁNÍ PERIFERÍÍ

Možné řešení – sonda (popis na předchozí stránce):



Níže je ukázka možného řešení pro základovou stanicí. Levý scénář ukazuje reakci na hodnotu poslanou ze sondy – nejprve nastaví proměnnou sonda na 1 (sonda je funkční). Poté vyhodnotí, zda obdržené číslo je 1 (poplach) nebo 0 (žádný poplach) a poplach případně spustí.

Pravý scénář pravidelně testuje, jestli není sonda rozbitá a zda funguje spojení. Odešle rádiem číslo a čeká 10 sekund na její reakci. Pokud do té doby neobdrží od sondy žádné číslo, hodnota proměnné sonda zůstane nastavena na 0 a spustí se poplach. Pokud číslo obdrží, v levém scénáři se mezitím nastaví hodnota proměnné **sonda** na 1 a poplach se v pravém scénáři nespustí.



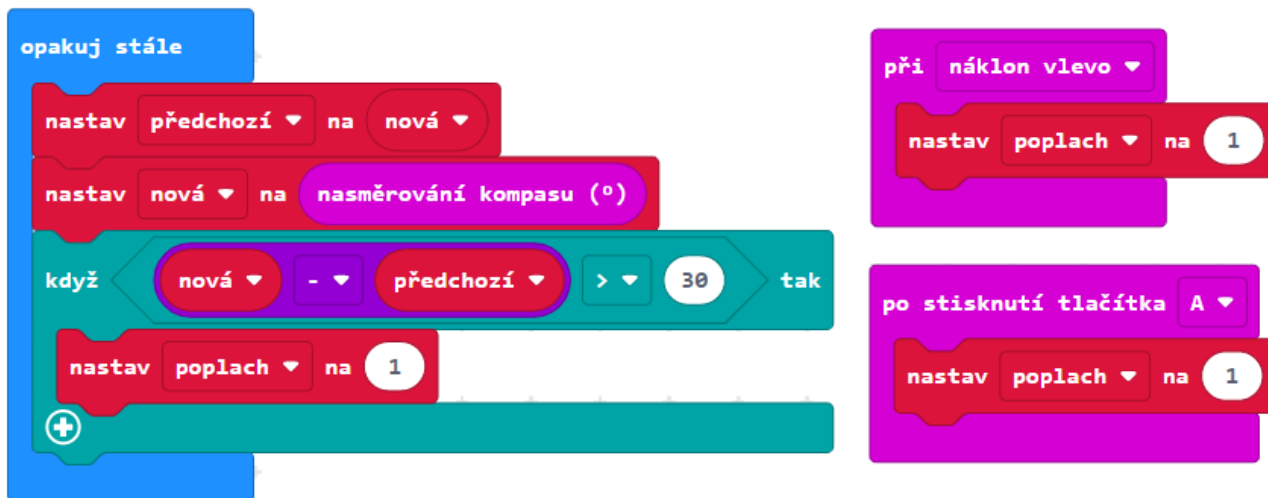
Jiným řešením kontroly sondy je, že základová stanice počítá, jestli dostává v nastaveném intervalu čísla rádiem (pro toto řešení neuvádíme program).

6 PŘIPOJOVÁNÍ PERIFÉRIÍ

Ukázka části řešení pro sondu – rozhodování podle změny polohy, jestli je, nebo není poplach. Levý scénář zapisuje naměřenou hodnotu do proměnné **nová** a porovnává ji s **předchozí** naměřenou hodnotou. Pokud jsou tyto dvě hodnoty dostatečně odlišné, poplach se spustí.

Pro přesnější určení je možné v podmínce použít absolutní hodnotu – pak vyjde rozdíl obou hodnot vždy kladný.

Levý scénář lze použít i pro změny intenzity světla a magnetického pole. Konkrétní hodnoty do podmínky je třeba odladit experimentálně.



Scénáře vpravo ukazují použití událostí (změna polohy, pokud bude např. micro:bit na víku truhly, která se otevírá, nebo když se poplach spustí po stisknutí nouzového tlačítka).

ZÁVĚR

Nyní jste již prošli celou učebnicí a měli byste micro:bit a MakeCode ovládat. Učebnice i kvůli některým úmyslným zjednodušením nemůže nahradit manuál, proto pokud vám něco chybělo anebo se chcete dozvědět více, navštivte weby micro:bit a MakeCode.

CO DÁL

Možná vás nyní napadne, co s micro:bitem dále. Zde bychom poradili zkusit žákům obstarat nějaký „microbití“ hardware.

- Lze zakoupit senzorové sady k micro:bitu, které obsahují řady čidel, servomotory, svítivé pásky LED diod apod. (micro:bit starter kit, Boson starter).
- Dají se koupit a smontovat například vozítka připravená pro micro:bit nebo simulace herní konzole (micro:bit :move, Yahboom).
- Můžete také micro:bitem ovládat chodícího minirobota Otto (lze koupit s tělem nebo i bez těla, které se vytiskne na 3D tiskárně).

Neuvádíme zde odkazy na konkrétní e-shop, věříme, že si svůj kus hardware spolu se žáky sami vyberete.

S micro:bity lze dělat celou řadu projektů na simulaci tzv. chytré domácnosti nebo ekologických zařízení. Mnoho nápadů najdete na internetu.

ODKAZY PRO DALŠÍ STUDIUM

<https://microbit.org/>

<https://makecode.microbit.org/>

<https://www.microbiti.cz/>

<http://robotika.sandofky.cz/otto-bot/>

B Příloha 2: Žákovské listy učebnice

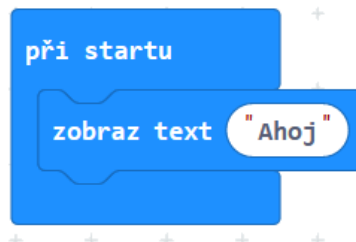
Uvedené žákovské listy jsou také dostupné na příloženém CD nebo online na portálu iMyšlení na adrese

<https://tinyurl.com/UcebniceMakeCode>

1. VÝPIS NA LED DISPLEJ

✓ 1. úkol

Poskládejte v prostředí *MakeCode* (<https://makecode.microbit.org/>) svůj první program:



Prohlédněte si jej vlevo na simulátoru.

Program poté uložte do souboru. Připojte k počítači micro:bit a podle pokynů od učitele soubor zkopírujte do micro:bitu (odešlete na micro:bit). Vyzkoušejte.

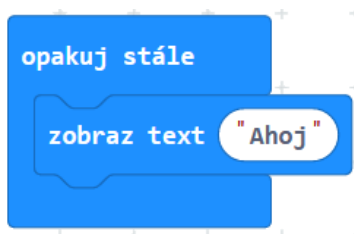
✓ 2. úkol

Napište své jméno na displej tak, že nebude přejíždět po displeji, ale jednotlivá písmena se budou střídat. V blocích `zobraz text` použijte jako parametr jen jedno písmeno.

Pokud si nevíte rady, zkuste nejprve zobrazit na displeji jenom jedno písmeno.

✓ 3. úkol

Poskládejte následující scénář. Jaký je rozdíl mezi ním a scénářem z 1. úkolu? Vysvětlete.



✓ 4. úkol

Sestavte program, který na micro:bitu postupně vypíše čísla od 0 do 9.

Použijte blok `opakuj 10 krát`.

✓ 5. úkol

Sestavte program, který na micro:bitu postupně vypíše čísla od 0 do 9. Nyní ale využijte blok `dokud`.

✓ 6. úkol

Pomocí opakování postupně vypište na displej sudá čísla menší než 8.

✓ 7. úkol

Vypište všechna lichá čísla větší než 0 a menší než 10 od největšího po nejmenší.

1. Výpis na LED displej

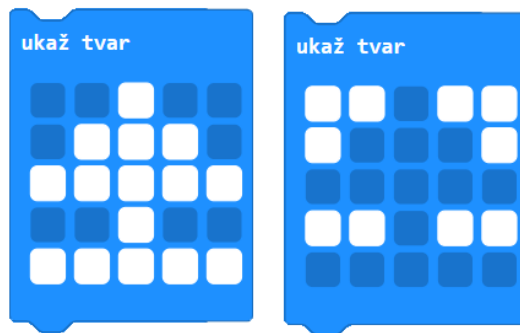
8. úkol

Sestavte program, který pomocí bloku `ukaz ikonu` zobrazí na displeji několik obrázků, a nakonec displej zhasne. Pro delší dobu zobrazení obrázků použijte blok `čekej`.

9. úkol

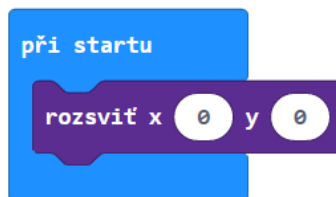
Vytvořte si vlastní obrázek pomocí bloku `ukaz tvar`. Dále vytvořte jeho inverzi (budou svítit diody, které jsou v původním obrázku rozsvíceny a naopak, viz obrázek). Vymyslete si svůj vlastní tvar.

Sestavte takový program, ve kterém se každý obrázek zobrazí třikrát.



10. úkol

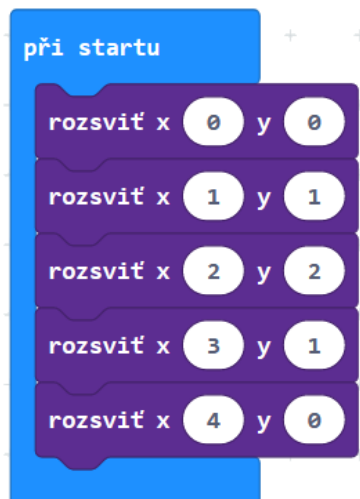
Vyzkoušejte, co dělá následující program:



Měňte čísla parametrů `x`, `y` a pozorujte.

Vysvětlete, co dělá blok `rozsviť`. Která čísla můžeme dosadit?

Nyní zkuste odhadnout, co se zobrazí na displeji po provedení tohoto kódu:



1. Výpis na LED displej

📌 11. úkol

Vykreslete na displeji svůj obrázek z 9. úkolu. Zajistěte, aby se diody rozsvěcovaly postupně, jedna po druhé.

📌 12. úkol

Postupně na displeji vykreslete první písmeno svého jména.

📌 13. úkol

Prozkoumejte blok `zhasni`. Sestav scénář, který tento blok použije.

📌 14. úkol

Pomocí bloku `ukáž tvar` vytvořte vlastní obrázek a ten pak postupně zhasínejte po jednotlivých bodech.

📌 15. úkol

Na obrázku vidíš scénář, který vykreslí animaci. Navrhni svůj vlastní scénář, který vykreslí podobnou animaci podle tvé volby.



1. Výpis na LED displej

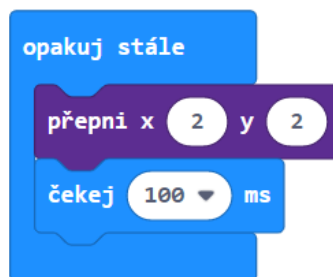
16. úkol

Jaký obrázek se rozsvítí podle tohoto scénáře?



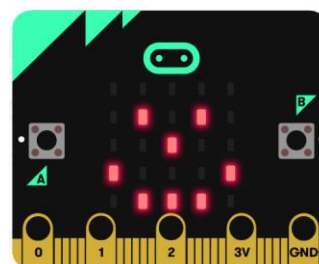
17. úkol

Vyzkoumejte, co dělá blok přepni. Zkuste určit, co dělá následující program:



18. úkol

Změňte smutný smajlík vytvořený blokem ukaž ikonu (obr. vlevo) na obrázek usmívajícího se smajlíka (obr. vpravo). Použijte blok přepni.



1. Výpis na LED displej

19. úkol

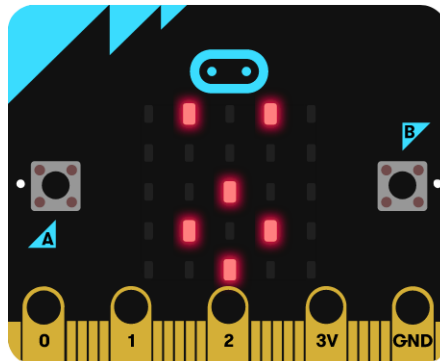
Honzka vytvářel v micro:bitu obrázek šipky dolů, ale jeho scénář vytvořil šipku doprava. Opravte jeho scénář:

```
při startu  
rozsviť x 0 y 2  
rozsviť x 1 y 2  
rozsviť x 2 y 2  
rozsviť x 3 y 2  
rozsviť x 3 y 1  
rozsviť x 3 y 3  
rozsviť x 4 y 2
```

20. úkol

Aktivita bez micro:bitu. Není dovoleno bloky přepisovat do počítače.

Vyberte ten scénář, který vykreslí obrázek na displeji. Svůj výběr zdůvodněte.



```
při startu  
rozsviť x 0 y 3  
rozsviť x 0 y 1  
rozsviť x 2 y 2  
rozsviť x 3 y 1  
rozsviť x 3 y 3  
rozsviť x 4 y 2
```

a)

```
při startu  
rozsviť x 3 y 0  
rozsviť x 1 y 0  
rozsviť x 2 y 2  
rozsviť x 1 y 3  
rozsviť x 3 y 3  
rozsviť x 2 y 4
```

b)

```
při startu  
rozsviť x 2 y 0  
rozsviť x 4 y 0  
rozsviť x 3 y 2  
rozsviť x 4 y 3  
rozsviť x 2 y 3  
rozsviť x 3 y 4
```

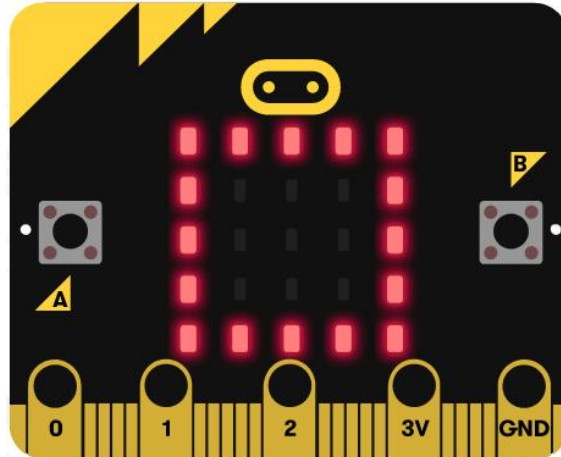
c)

1. Výpis na LED displej

21. úkol

Vytvořte scénář, který postupně rozsvítí krajní diody. Měly by se rozsvěcet po směru hodinových ručiček vždy po 200 ms.

Výsledný obrázek na displeji:



2. PRÁCE S TLAČÍTKY

✓ 1. úkol

Sestavte a prozkoumejte tyto scénáře. Popište, co se děje, když stiskneme tlačítka A a B na micro:bitu.



Umístěte dovnitř bloků po stisknutí i další bloky, například zobraz text.

✓ 2. úkol

Přidejte do programu další blok po stisknutí tlačítka. V jeho rozbalené nabídce vyberte třetí možnost A+B. Vyzkoušejte, jak tento blok spustit.

K čemu se tento blok hodí?

✓ 3. úkol

Vyzkoušejte, co dělá blok při zatřesení. Sestavte další scénář, který zobrazí jinou ikonu.

Kdy se tato událost dá využít?

✓ 4. úkol

Vytvořte program, který bude neustále na displeji zobrazovat, kolikrát bylo stisknuto tlačítko A.

Před vytvářením programu odpovězte na následující otázky:

1. Jak si bude micro:bit pamatovat, kolikrát se tlačítko stisklo?
2. Jak zařídit, aby na displeji svítila zapamatovaná hodnota?
3. Jak naprogramovat, co se stane po stisknutí tlačítka?

Nakonec přidejte scénář, který po stisknutí tlačítka B naopak jedničku odečte.

K čemu by se takto naprogramovaný micro:bit dal použít?

✓ 5. úkol

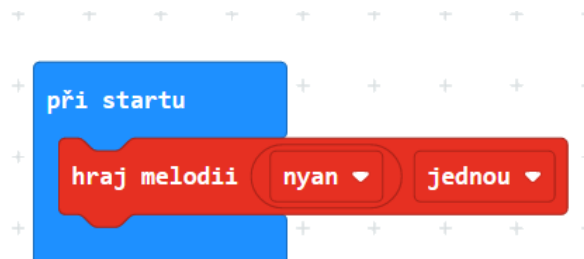
Naprogramujte postřehovou hru **Chytání písmenek**. Na micro:bitu se bude střídavě náhodně zobrazovat a zhasínat písmeno A nebo B a hráč bude muset stisknout odpovídající tlačítko, než písmeno zmizí z displeje. Za „chycené“ písmenko se hráči přičte bod. Hra skončí po daném počtu opakování. Dobu zobrazení a čekání na stisk lze při ladění hry nastavit tak, aby se hra dobře hrála (ani pomalé, ani rychlé). Na závěr může být vyhodnocení např. počtem úspěšných pokusů.

Pro volbu A nebo B použijte blok náhodné číslo ze sekce **Matematika**.

3. HUDBA

✓ 1. úkol

Zapojte podle pokynů učitele do micro:bitu sluchátka. Poté do micro:bitu nahrajte následující program:



Prozkoumejte, co tento blok dělá. Jaký je rozdíl mezi možnostmi jednou, stále dokola, jednou na pozadí a stále dokola na pozadí v posledním sloupci?

✓ 2. úkol

Vytvořte takový program, který po stisknutí tlačítka A zobrazí na displeji jednu z předpřipravených ikon a přehraje nějakou melodii.

Upravte program tak, aby se po stisknutí tlačítka B zobrazila jiná ikona a přehrála jiná melodie.

✓ 3. úkol

Najděte v nabídce bloků blok `přehraj melodii`. Sestavte si pomocí něj vlastní melodii a přehrajte.

✓ 4. úkol

Sestavte tyto dva scénáře:



Co dělá blok `poté, co`? Měňte podmínky a zjistěte, jak se mění chování micro:bitu.

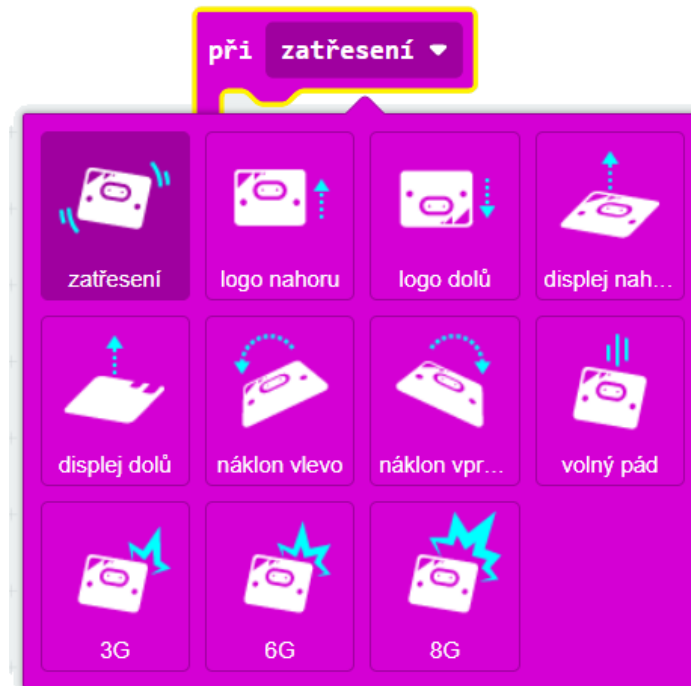
✓ 5. úkol

Se znalostmi z předchozích úkolů sestavte program, který přehraje kratší známou melodii. Například koledu *Rolničky* nebo 1. sloku písničky *Ovčáci čtveráci*.

4. POLOHA

1. úkol

Prohlédněte si blok při zatřesení ze sekce **Vstup** – rozbalte jeho nabídku, prohlédněte si a zkuste odhadnout, co jednotlivé události v této nabídce znamenají.



2. úkol

Sestavte program, který při naklonění micro:bitu vlevo nebo vpravo ukáže různé ikony.

3. úkol

Sestavte program, který po naklonění micro:bitu vlevo nebo vpravo ukáže šipku, která bude ukazovat nahoru, ke stropu učebny.

4. úkol

Předchozí program z 3. úkolu má nevýhodu, že šipka zůstává na displeji i tehdy, když se micro:bit vrátí do vodorovné polohy. Přidejte událost, která na displeji zobrazí smajlíka, když micro:bit leží vodorovně na stole.

5. úkol

Vytvořte program, který zobrazí veselý smajlík, když se micro:bit postaví na stůl logem nahoru (tedy dírkami dolů), a smutný smajlík, když se převrátí vzhůru nohama.

6. úkol

Otestujte další události.

Vymyslete, k čemu praktickému by se některá z událostí dala využít, a sestavte pro to program.

7. úkol

Vyzkoušejte následující program. Měňte dolní i horní hranici. Jak se chování scénáře změní?



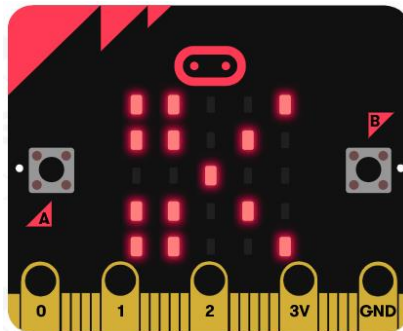
Zkoušejte opakovaně.

K čemu by se takový program mohl použít?

8. úkol

Vytvořte hru Kámen, nůžky, papír. Micro:bit bude při zatřesení náhodně zobrazovat písmeno K pro volbu kámen, N pro nůžky a P pro papír.

Pro nastavení více možností v bloku *když stisknete* na tomto bloku ikonu ⊕.



Zahrajte si hru „člověk proti micro:bitu“. Jednou rukou nastavte kámen, nůžky nebo papír, druhou zatřeste micro:bitem. Vyhrál micro:bit nebo Vy?

9. úkol

Vytvořte program, který po zmáčknutí tlačítka A napíše náhodné číslo od 0 do 9. K čemu se takový program může hodit?

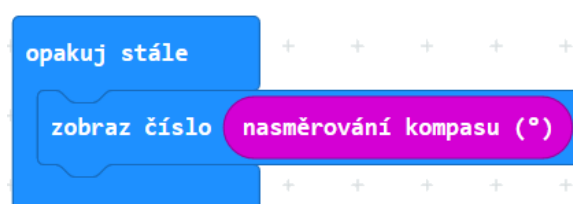
10. úkol

Vytvořte program, který po zmáčknutí tlačítka A vygeneruje náhodné PIN, PSČ, telefonní číslo nebo číselné heslo. Zobrazovat jej bude po číslicích a na konci displej zhasne.

11. úkol

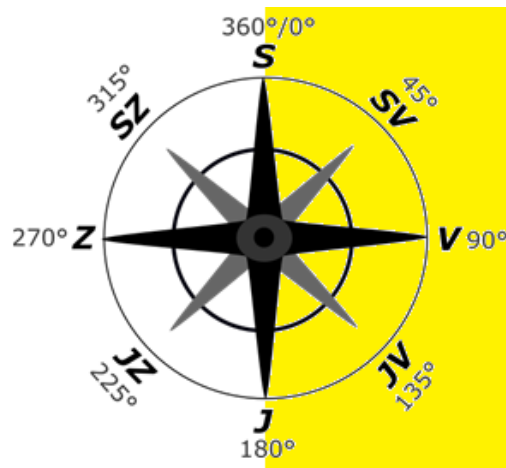
Naučte se s pomocí učitele používat kompas.

Zjistěte, co se a v jakých hodnotách ukazuje na micro:bitu po spuštění následujícího programu.



📌 12. úkol

Následující obrázek ukazuje hodnoty čísel, které kompas micro:bitu udává pro světové strany.



Určete, jaké hodnoty bude micro:bit ukazovat, když bude natočen více na východ než na západ (žlutě podbarveno).

📌 13. úkol

Honza chce vytvořit program, který by na micro:bitu písmeny V, Z ukazoval, jestli jde Honza na západ nebo na východ.

Vytvořte takový program.

📌 14. úkol

Upravte program z předchozího úkolu tak, aby micro:bit ukazoval do čtyř světových stran S, V, J, Z.

📌 15. úkol

Prozkoumejte, co dělá blok `intenzita světla` ze sekce **Vstup**.

Vytvořte program, v němž bude displej stále zobrazovat hodnotu intenzity světla.

Micro:bit přenášejte a natáčejte. Jakých hodnot nabývá změřená intenzita světla? Snažte se docílit co nejnižší a co nejvyšší hodnoty.

📌 16. úkol

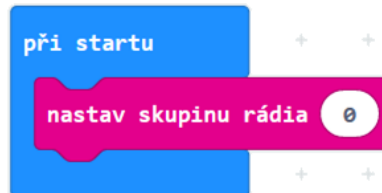
Se znalostmi z předchozího úkolu vytvořte program, který rozezná noc a den.

Pomocí pokusů sami rozhodněte, jaká hraniční hodnota intenzity světla bude znamenat změnu dne v noc.

5. SÍŤ

1. úkol

Podle pokynů učitele si označte jeden micro:bit jako *přijímač*, a druhý jako *vysílač*. Oba budou potřebovat tento scénář. Do obou sestavte scénář, v bloku nastav skupinu rádia zadejte stejné číslo, které vám přidělí učitel:



Každá skupina micro:bitů ve třídě musí mít jiné číslo.

Proveďte zkoušku spojení zadáním bloků pro vysílání a příjem signálu.

Na micro:bitu označeném *vysílač* sestavte:



Pro micro:bit *přijímač* sestavte scénář:



Sledujte a řekněte, co se stalo po spuštění programu.

2. úkol

Přidejte další scénáře, které zařídí, aby se po stisknutí tlačítka A na druhém micro:bitu přenesl signál na první micro:bit a zobrazilo se písmeno na displeji.

Přidejte scénář, který reaguje na zatřesení a zhasne displej.

3. úkol

Nyní vytvořte program, který z micro:bitu udělá počítadlo „na dálku“.

Po stisknutí tlačítka A *vysílač* vyšle číslo. *Přijímač* po přijetí čísla přičte jedničku a zobrazí počet stisknutí na displeji.

✓ 4. úkol

Do *vysílače* přidejte takový scénář, díky kterému bude micro:bit vysílat při stisknutí tlačítek A, B dvě různé zprávy.

✓ 5. úkol

Vysvětlete, k čemu slouží následující scénář.



Co znamená proměnná **receivedNumber**? Jakou hodnotu obsahuje, když *vysílač* pošle číslo 1?

Scénář ve svém programu upravte tak, aby se na displeji zobrazila hodnota této proměnné **receivedNumber**.

✓ 6. úkol

Upravte program tak, aby *vysílač* vysílal náhodné číslo.

Po vyzvání učitele změňte skupinu rádia na obou micro:bitech na číslo 0.

Vyzkoušejte přes *vysílač* poslat zprávu. Co se děje?

✓ 7. úkol

Nastavte na obou micro:bitech skupinu rádia na původní číslo.

Upravte kód pro *přijímač* tak, aby na displej vypsal, které tlačítko bylo na vysílači stisknuto. Tedy zobrazí „A“, pokud je na *vysílači* stisknuto tlačítko A, a „B“ při stisku tlačítka B.

✓ 8. úkol

Najděte na internetu tabulku s Morseovou abecedou.

Vytvořte programy pro *vysílač* a *přijímač*, pomocí kterého půjde posílat zprávy zašifrované v Morseově abecedě.

Program otestujte posláním jednoho zašifrovaného písmene.

✓ 9. úkol

Učitel vám rozdá papírky se zprávou, kterou bude mít za úkol *vysílač* poslat *přijímači*.

Pokud během posílání zprávy narazíte na problém, programy upravte tak, aby k němu nedocházelo.

✓ 10. úkol

Nyní zakódujte a pošlete jednoslovnou zprávu.

Pokud během posílání zprávy narazíte na problém, programy upravte tak, aby k němu nedocházelo.

Poté pošlete víceslovnou zprávu. Znovu opravte všechny problémy, na které narazíte.

11. úkol

V této úloze budete mít za úkol postupně sestavit systém pro řízení ankety.

Při hlasování budete mít jeden centrální micro:bit a několik hlasovacích micro:bitů. Centrální micro:bit bude shromažďovat údaje (hlasy), poslané z hlasovacích micro:bitů. Hlasovací micro:bity budou mít všechny stejný program. Budou hlasovat tak, že se buď stiskne tlačítko A (jsem pro návrh) nebo B (jsem proti návrhu). Centrální micro:bit hlasy zaznamená, sečte a na požádání zobrazí.

Ještě, než se pustíte do programování, zamyslete se nad následující problémy:

1. Jak hlasovací micro:bit pozná, že už se smí hlasovat?
2. Jak zařídit, aby každý micro:bit mohl hlasovat pouze jednou?
3. Jaké různé signály se budou posílat?
4. Co má vlastně centrální micro:bit dělat?
5. Jak si centrální micro:bit bude pamatovat, kolik hlasů má každá z možností pro/proti?
6. Jak centrální micro:bit zobrazí výsledky hlasování?

Jejich řešení vás posune směrem k řešení úkolu.

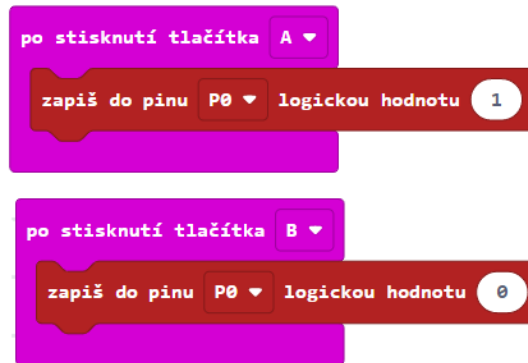
Program sestavte buď ve skupině, nebo s pomocí učitele, a náležitě otestujte.

6. PŘIPOJOVÁNÍ PERIFÉRIÍ

1. úkol

Podle instrukcí učitele zapojte dva z pinů na tříbarevné diodě. Ten nejdelší připojte k pinu GND na micro:bitu a libovolný další k pinu 0. Ostatní nepřipojujte.

Vyzkoušejte následující scénáře:



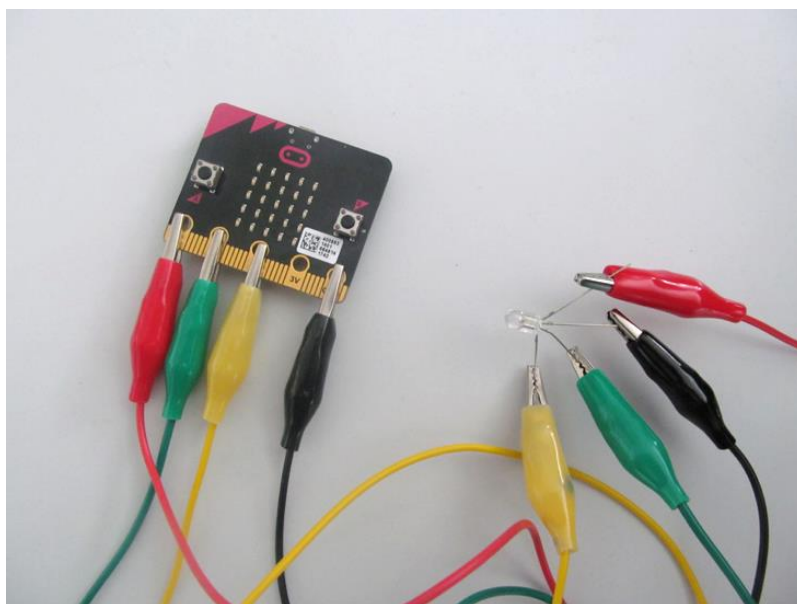
Co se stane s diodou po zmáčknutí tlačítek A a B?

Nejdelší pin na diodě nechejte, jak je, a přepojte krokosvorku u druhého pinu k jinému pinu na diodě. Spustěte program a sledujte, co se změnilo.

Nyní sestavte scénář, v němž dioda několikrát zabliká.

2. úkol

Zapojte tříbarevnou diodu k micro:bitu podle obrázku. Zem (GND) zapojte na nejdelší pin tříbarevné diody. Ostatní zapojení je doporučeno. Pin 0 zapojte k samostatnému pinu diody (červená), který je na jedné straně od nejdelšího pinu diody. Oba piny z druhé strany diody zapojte tak, že pin blíže ke středu (zelená) připojíte k pinu 1 a poslední pin (modrá) k pinu 2. Viz obrázek.

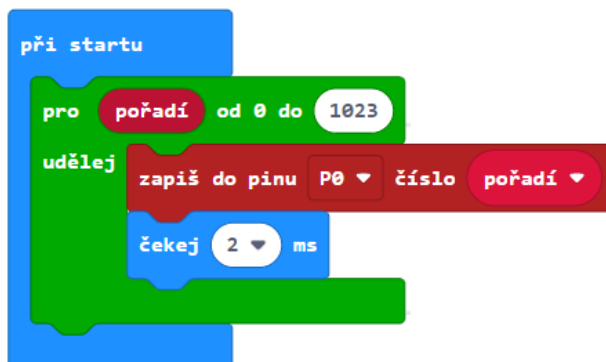


Sestavte scénář, který nejprve rozsvítí diodu na pin 0 a po jedné sekundě ji zhasne. Poté rozsvítí diodu na pin 1 (a opět ji po sekundě zhasne) a nakonec rozsvítí diodu na pin 2 a opět ji zhasne.

6. Připojování periférií

3. úkol

Vyzkoušejte následující scénář:



Uřčete, jaký je rozdíl mezi tímto zapojením a mezi zapojením z 1. úkolu?

4. úkol

Upravte scénář z předchozího úkolu, aby dioda po rozsvícení opět postupně vyhasínala.

5. úkol

Vytvořte program, který z diody udělá magickou lampu. Lampa bude náhodně střídat tři barvy. Nejdříve jednu vybere a rozsvítí, po chvíli zhasne a rozsvítí jinou barvu.

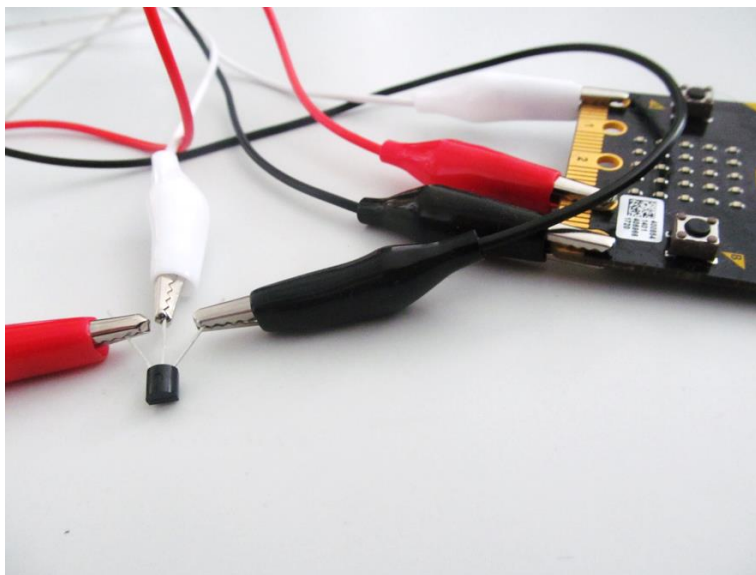
Pokud si nevíte rady, zamyslete se nad následujícími otázkami:

1. Jak vybrat náhodně jednu barvu?
2. Jak zajistit, aby program nevybral dvakrát za sebou stejnou barvu?
3. Jak program vybranou barvu rozsvítí?
4. Jak zajistit zhasnutí vybrané barvy před rozsvícením nové?
5. Mezi rozsvícením a zhasínáním diody musíme dát čas na čekání. Jak to udělat, aby tento čas nebyl pořád stejný?

Nyní vyrobte skutečnou lampu, například jenom jako váleček ze čtvrtky, kde jednotlivé piny prostrčíte čtvrtkou ven. Důležité je, aby se jednotlivé konektory navzájem nedotýkaly.

Program můžete rozšířit přidáním dalších barev zapnutím dvou nebo více pinů najednou.

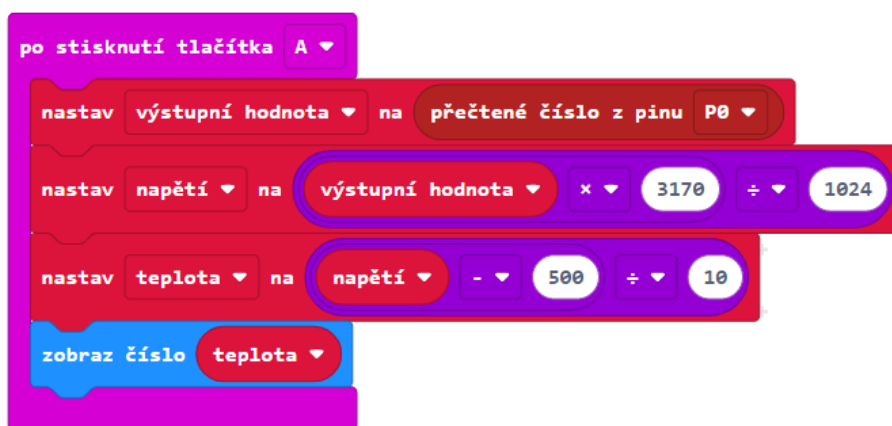
6. úkol



Podle obrázku a instrukcí od učitele propojte s micro:bitem teplotní čidlo.

Sestavte scénář, který podle vzorců přepočítá přečtenou hodnotu z teplotního čidla na naměřenou teplotu. Podle druhu použitého čidla mohou být hodnoty ve vzorcích odlišné – poraďte se s učitelem.

Sledujte, jaký údaj se zobrazí na displeji.



Své měření porovnejte se spolužáky.

Scénář upravte tak, že bude teplotu měřit automaticky po 10 sekundách a trvale ji zobrazovat.

7. úkol

Se zkušenostmi z předchozího úkolu vytvořte teploměr se vzdálenou sondou. Využijte dva micro:bity. Jeden bude teploměrem: bude se ovládat tlačítka a bude zobrazovat teplotu. Druhý bude sondou, bude připojen k teplotnímu čidlu a bude odesílat rádiem naměřené hodnoty do micro:bitu – teploměru.

Sestavte scénáře pro teploměr i pro sondu. Micro:bit - sondu umístěte za okno a na druhém micro:bitu uvnitř místnosti sledujte venkovní teplotu.

8. úkol

Signalizátor teploty na dálnici. Potřebujeme teplotní čidlo a trojbarevnou diodu.

Ze dvou micro:bitů sestavte základovou stanici a teplotní sondu. Teplotní sonda bude měřit teplotu pomocí externího čidla a pomocí rádia bude výsledky v určitém intervalu (např. 10 sekund) předávat základové stanici. Základová stanice bude zobrazovat naposledy obdrženu teplotu a podle naměřené teploty diodou signalizovat stupeň zahřátí vozovky. (např. méně než 3 °C – modrá, nebezpečí náledí, 3 °C až 20 °C – zelená, vyšší - červená).

Signalizátor teploty můžeme vyzkoušet s jinými hraničními teplotami, kterých můžeme ve třídě dosáhnout (např. 16°C, 23°C). Teplotní čidlo je možné pro vyzkoušení ohřívat a chladit, ale ne ve vodě – hrozí zkrat.

9. úkol

Závěrečný projekt - zabezpečovací zařízení.

Pomocí dvou micro:bitů sestavte zabezpečovací zařízení, které bude hlásit změnu polohy micro:bitu, změnu intenzity světla, teploty, magnetického pole apod.

Jeden micro:bit bude fungovat jako čidlo a druhý jako základová stanice.

Vyberte si z možností, na co bude čidlo reagovat:

- Teplota překročila určitou mez.
- Reaguje na změnu polohy. Např. micro:bit je umístěn na dveřích, které se otevřou, nebo je umístěn v hlídaném předmětu, který je zvednut (otočen).
- Reaguje na významnou změnu intenzity světla (rozsvíceno, zhasnuto) pomocí bloku `intenzita světla`.
- Reaguje na významnou změnu magnetického pole pomocí bloku `magnetická síla` v sekci *Vstupy*. Např. micro:bit je umístěn na dveřích, které se vzdálí od kovového rámu, nebo se od něj vzdálí magnet, umístěný v hlídaném předmětu.
- Je stisknuto tlačítko – nouzový signál.

Ve všech těchto případech čidlo vyhlásí poplach (pošle signál základové stanici). Ta podle vybavení rozsvítí diody na displeji, zabliká trojbarevnou diodou nebo vyšle zvukový signál. Stiskem tlačítka na základové stanici je možné poplach odvolat.

Čidlo by se mělo pravidelně hlásit, a pokud tak neučiní (např. při poruše), měl by poplach být rovněž vyhlášen.

Můžete sestavit zabezpečovací zařízení, které kombinuje více prvků ochrany (např. hlídá teplotu a ještě umožňuje spustit poplach stisknutím tlačítka).