



Pedagogická  
fakulta  
Faculty  
of Education

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

Jihočeská univerzita v Českých Budějovicích  
Pedagogická fakulta  
Katedra informatiky

Bakalářská práce

Projektová výuka s využitím BBC Micro:bit  
Project Based Learning with the use of BBC  
Micro:bit

Vypracoval: Štefl Michael  
Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.  
Odborný konzultant: Mgr. Jan Pršala

České Budějovice 2024

# JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta  
Akademický rok: 2021/2022

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Michael ŠTEFL  
Osobní číslo: P20328  
Studijní program: B0114A300110 Oborové studium se zaměřením na vzdělávání na 2. stupni základní školy  
Specializace: Pedagogicko-psychologický základ a předměty učitelské propedeutiky  
Anglický jazyk se zaměřením na vzdělávání na 2. stupni ZŠ  
Informační technologie se zaměřením na vzdělávání na 2. stupni ZŠ  
Téma práce: Projektová výuka s využitím BBC Micro:bit  
Zadávací katedra: Katedra informatiky

### Zásady pro vypracování

Cílem bakalářské práce je vytvořit sadu úloh pro základní školu, které budou využívat programovatelnou desku BBC micro:bit rozšířenou o sadu při daných elektronických vstupů a výstupů. Úlohy budou určeny pro projektovou výuku a budou sloužit jako závěrečné opakování práce s micro:bitem. K programování desky bude použito blokové prostředí. V teoretické části student stručně popíše programovatelnou desku micro:bit a zvolené rozšíření. Dále vybere a zhodnotí možnosti různých blokových prostředí pro práci s periferiemi. Popíše také zásady projektové výuky. Nakonec najde dostupné projektové úlohy, které využívají micro:bit, na internetu, a popíše je. V praktické části student vymyslí a navrhne několik úloh – projektů, ve kterých se micro:bit zapojí společně s další periferií (LED dioda, větráček, mikrofon, ...), určených pro 2. stupeň ZŠ. Součástí úloh budou náměty na vzhled a postupu vytvoření výrobků, do kterých bude micro:bit a periferiemi vsazen, aby tvořily funkční celek. Tyto projekty následně otestuje ve výuce nebo na kroužku robotiky a podle zpětné vazby z tohoto ověření sadu upraví.

Rozsah pracovní zprávy: 40  
Rozsah grafických prací: interaktivní modely  
Forma zpracování bakalářské práce: tištěná

### Seznam doporučené literatury:

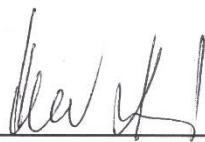
1. Juříček, J. (n. d.). Retrieved from: <https://didatech.cz/>
2. Juříček, J. (n. d.). Retrieved from: <https://ucimesroboty.cz/index.php/category/bbc-microbit/>
3. Kubínová, M. (2005). Projekty ve vyučování. Metodický portál RVP. Retrieved from: <https://clanky.rvp.cz/clanek/c/z/334/PROJEKTY-VE-VYUCOVANI.html>
4. Loužecká, I. (2020). Retrieved from: <https://www.veskole.cz/clanky/projektova-vyuka-nejen-v-soucasne-situaci>
5. Mazáčová, N. (2007). Retrieved from: <https://clanky.rvp.cz/clanek/c/z/1288/MOZNOSTI-A-MEZE-PROJEKTOVE-VYUKY-V-SOUCASNE-SKOLE.html>
6. Micro:bit. (n. d.). Retrieved from: <https://microbit.org>
7. Pech, J., Pršala, J., Vaníček, J., & Novák, M. (2021). Robotika pro základní školy: programujeme micro:bit pomocí Makecode.
8. Seneviratne, P. (2019). BBC micro:bit Recipes.
9. Tomková, A., Kašová, J., & Dvořáková, M. (2009). Učíme v projektech. (p. 176).
10. Tooley, M. (2017). Getting Started with the BBC Micro:Bit. (p. 128).
11. Zormanová, L. (2012). Retrieved from: <https://clankv.rvp.cz/clanek/c/Z/14983/projektova->

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

Konzultant bakalářské práce: **Mgr. Jan Pršala**  
Katedra informatiky

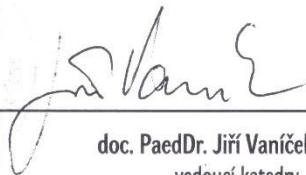
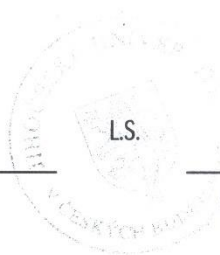
Datum zadání bakalářské práce: **4. dubna 2022**

Termín odevzdání bakalářské práce: **30. dubna 2023**



---

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



---

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

## **Prohlášení**

Prohlašuji, že svoji bakalářskou 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 bakalářské, 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 19. dubna 2024.

Michael Štefl

## **Anotace**

Tato práce se zabývá využitím micro:bitu v projektové výuce. Teoretická část práce se zabývá popisem micro:bitu, projektové výuky, porovnáním blokových prostředí a zvolením toho nejlepšího. Zároveň popisuje možné zdroje pro projektové úlohy s micro:bitem, které fungovaly jako inspirace pro úlohy v praktické části. Praktická část se zabývá tvorbou úloh, jejich popsáním a vyzkoušením ve výuce. Na závěr jsou provedeny úpravy na základě zpětné vazby.

## **Klíčová slova**

Micro:bit, Microsoft Makecode, blokové prostředí, robotika, programování

## **Abstract**

This thesis deals with the use of micro:bit in project-based learning. The theoretical part of the thesis deals with the description of micro:bit, project-based learning, comparing block environments and choosing the best one. It also describes possible resources for project tasks with micro:bit, which acted as inspiration for the tasks in the practical part. The practical part deals with creating tasks, describing them and testing them in the classroom. Finally, adjustments are made based on feedback.

## **Keywords**

Micro:bit, Microsoft Makecode, block environment, robotics, programming

## **Poděkování**

Chtěl bych poděkovat především Mgr. Janu Pršalovi za vedení, rady, názory, zpětnou vazbu a velkou dávku trpělivosti. Dále děkuji panu doc. PaedDr. Jiřímu Vaníčkovi, Ph. D. za umožnění psaní této práce.

Dále bych chtěl poděkovat své přítelkyni, za podporu a obrovské pochopení při psaní této práce. A v neposlední řadě chci taky poděkovat svým přátelům a rodině za veškerou podporu, které se mi během psaní této práce dostalo.

## Obsah

<b>1. Úvod</b> .....	<b>10</b>
1.1. Cíle práce .....	10
1.2. Metody práce.....	10
<b>2. Micro:bit</b> .....	<b>12</b>
2.1. Historie micro:bitu .....	13
<b>3. Bloková prostředí pro programování destičky micro:bit</b> .....	<b>15</b>
3.1. Scratch.....	15
3.2. Python editor .....	16
3.3. Microsoft Makecode .....	17
3.4. Vittascience .....	18
3.5. Shrnutí.....	19
<b>4. Úlohy projektové výuky dostupné na internetu</b> .....	<b>21</b>
4.1. Makecode Microbit .....	21
4.2. Microbiti .....	22
4.3. Učíme s roboty .....	22
4.4. Oficiální stránka Microbit .....	24
<b>5. Projektová výuka</b> .....	<b>26</b>
5.1. Postup při projektovém vyučování.....	26
5.2. Pozitiva a negativa projektové výuky .....	27
5.3. Shrnutí.....	28
<b>6. Projekty</b> .....	<b>29</b>
6.1. Automatická lampa .....	29
6.2. Teploměr .....	34
6.3. Alarm na dveře.....	35
6.4. Zabezpečení proti krádeži .....	38
6.5. Měřič hluku.....	40
<b>7. Odkoušení v praxi</b> .....	<b>42</b>
7.1. Zpětná vazba.....	46
7.2. Úpravy na základě zpětné vazby .....	47
7.2.1. Úvod .....	47
7.2.2. Automatická lampa .....	47
7.2.3. Teploměr .....	48
7.2.4. Alarm na dveře .....	49
7.2.5. Zabezpečení proti krádeži.....	51



7.2.6. Shrnutí .....	52
<b>8. Závěr.....</b>	<b>53</b>
<b>Seznam použité literatury a zdrojů .....</b>	<b>54</b>
<b>Použité obrázky .....</b>	<b>56</b>
<b>Přílohy.....</b>	<b>57</b>
<b>Příloha 1 .....</b>	<b>57</b>

# 1. Úvod

Tato práce je zaměřena na tvorbu projektových úloh pro micro:bit. Obsahuje úlohy, které využívají micro:bit a jeho periferie. Zároveň u toho žáci mohou pracovat samostatně vzhledem k tomu, že Micro:bit má přívětivé pracovní prostředí.

Toto téma jsem si zvolil, protože mě baví informatika, ale zároveň jsem chtěl něco, co bude mít i pedagogickou část. Vzhledem k tomu, že jsem většinou pracoval spíše se softwarem než s hardwarem, tak to pro mě byla možnost se něčemu novému sám naučit a získat užitečnou praxi pro mou budoucí výuku. Zároveň mi jednodeskové počítače přijdou zajímavé a chci zjistit, čeho jsou schopné a jak se dají prakticky využít.

Výstupem této práce jsou nové poznatky ohledně toho, jak u žáků provést projektovou výuku, materiály pro případné projekty, nové poznatky, co se projektové výuky robotiky a micro:bitu týče, a zároveň názory žáků, co fungovalo a co ne. Nejdůležitější výstup celé bakalářské práce jsou ovšem samotné úlohy pro práci s micro:bitem.

## 1.1. Cíle práce

Cílem bakalářské práce bylo vytvořit sadu úloh, které budou využívat programovatelnou desku BBC micro:bit a jeho periferie. Úlohy jsou určeny pro projektovou výuku a budou sloužit jako závěrečné opakování práce s micro:bitem. Teoretická část této práce se zabývá popsáním micro:bitu a jeho historií a fungováním. Dále se zabývá blokovým programováním, prostředími pro něj a porovnává je. Byly popsány úlohy dostupné na internetu. Nakonec se zabývá projektovou výukou, jejím využitím, jejími způsoby a pozitivy i negativy, které přináší.

Praktická část obsahuje zadání úloh pro projektovou výuku pokročilejších žáků na základní škole. Každá úloha obsahuje zadání, rozšířený popis pro učitele, ukázkové řešení v kódu a návrh na zapojení a konstrukci. Následně je popsán průběh testování těchto úloh, jehož cílem bylo získat zpětnou vazbu. Tato zpětná vazba je shrnutá v poslední kapitole, společně s úpravami pro úlohy. To vše bylo zpracováno a zhodnoceno v této bakalářské práci.

## 1.2. Metody práce

Jako první jsem si vyzkoušel sám micro:bit a naučil se s ním pracovat. Zjistil jsem, jak připojit periferie a ty poté nastavit. Zároveň jsem zde popsal jeho historii a rozdíly mezi dostupnými verzemi. Dále jsem z literatury nastudoval projektovou výuku, jak funguje, jak jí správně připravit, čemu se vyvarovat a co je efektivní. Dále jsem si při tvorbě teoretické části vybral

a vyzkoušel různá bloková programovací prostředí, ty stručně popsal, zhodnotil a vybral z nich prostředí, které nejvíce vyhovovalo mým úmyslům a se kterým se pracovalo nejlépe. Tímto prostředím byl zvolen Microsoft Makecode. Popsal jsem úlohy již dostupné na internetu a zhodnotil jejich pozitiva a negativa. V poslední teoretické kapitole jsem se zaměřil na samotný koncept projektové výuky, popsal její výhody a nevýhody a jak postupovat při jejím provedení.

V praktické části jsem vymyslel a navrhl několik úloh – projektů, ve kterých se micro:bit zapojí společně s další periférií (LED dioda, větráček, bzučák, ...), určených pro 2. stupeň ZŠ. K těmto úlohám jsem vymyslel zadání, popis, možné řešení a zapojení společně s konstrukcí. Tyto projekty jsem následně otestoval ve výuce. Poté jsem se žáků ptal, co jim vyhovovalo, a zjistil od nich co nejpřesnější zpětnou vazbu. Pomocí zpětné vazby a mého pozorování jsem následně úlohy upravil, aby žákům i učitelům lépe vyhovovaly.

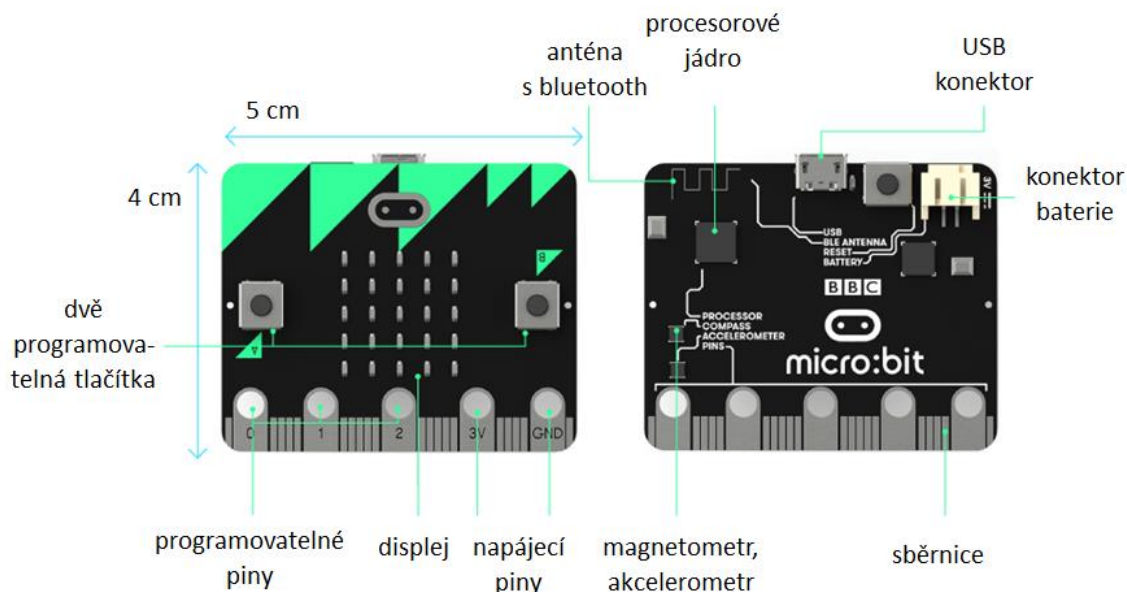
## 2. Micro:bit

Micro:bit je programovatelná destička vyvinuta pro výuku informačních technologií. Lze ho programovat za pomoci několika jazyků – blokový Microsoft Makecode nebo textový Javascript a MicroPython [1].

Možnost programování blokově či pomocí kódu umožňuje jednoduše žákům pochopit základy a ty poté rozvést dále.

Samotná destička micro:bitu obsahuje:

- **Displej:** Micro:bit obsahuje pět x pět LED diod (5x5 LED mřížka), které umožňují zobrazování textu, symbolů a animací. Tento displej umožňuje žákům interaktivní projekty.
- **Tlačítka:** Dvě tlačítka (A a B) umožňují interakci s programy a projekty. Tyto tlačítka lze snadno programovat k různým funkcím, což rozšiřuje možnosti tvorby projektů.
- **Senzory:** Micro:bit obsahuje integrované senzory, jako je akcelerometr (pohyb), magnetometr (orientace), senzory teploty, měřič intenzity světla. Tyto senzory jsou společně s tlačítky klíčové pro tvorbu interaktivních projektů, jako jsou například hry nebo interaktivní systémy.
- **Rozhraní:** Micro:bit disponuje vstupně/výstupními piny (GPIO piny), které umožňují připojení dalších periférií a senzorů, jako jsou například snímače teploty, reproduktory nebo motory. Tato rozšiřitelnost dává žákům možnost experimentovat a vytvářet rozmanité projekty.
- **Piny:** Možnost zapojení krokosvorek či krokodýlů, zapojení dalších periférií.



Obrázek 1: schéma Micro:bit

K destičce lze dokoupit další příslušenství jako třeba držák na baterie, kabelosvorky nebo rozšiřující moduly [2] [3].

## 2.1. Historie micro:bitu

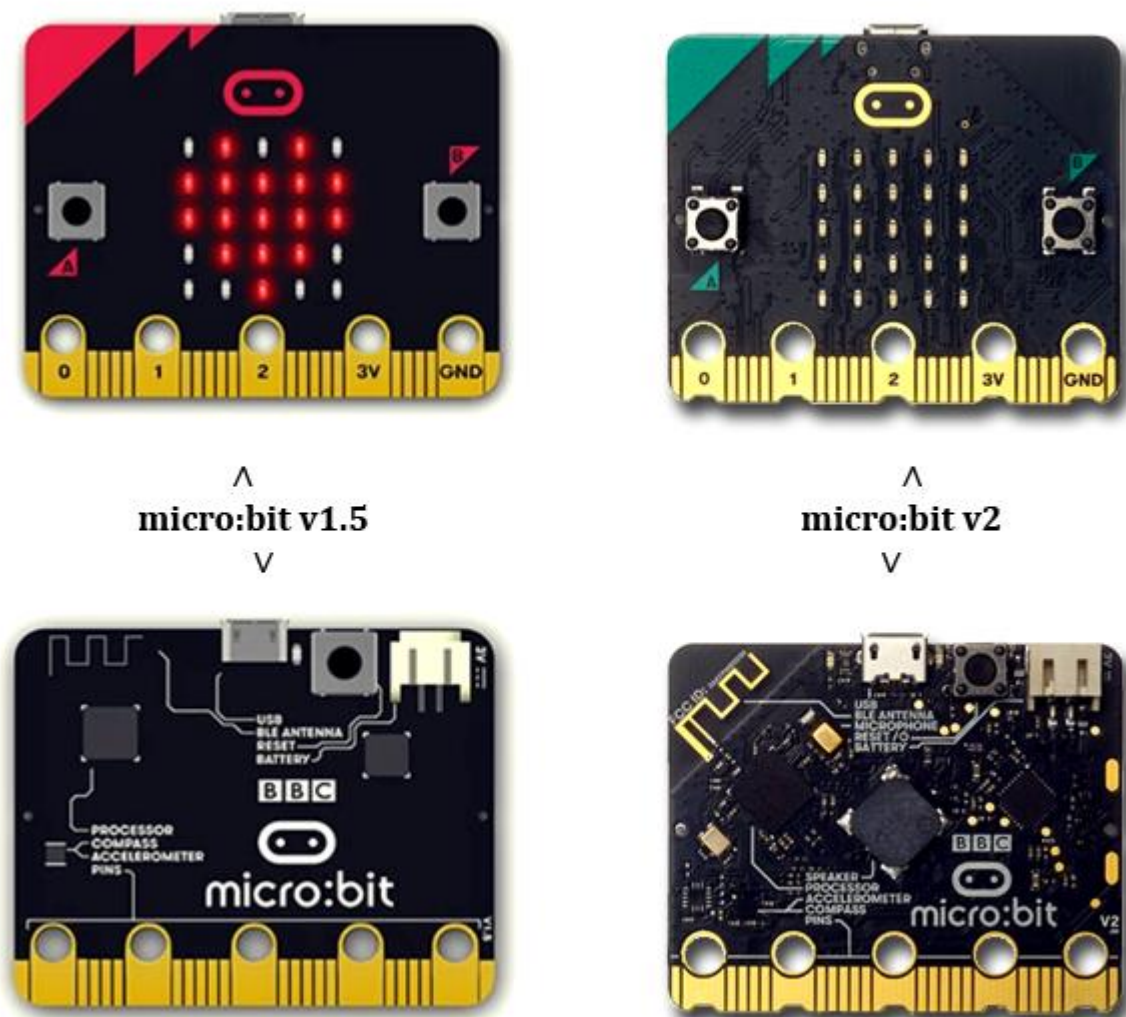
Micro:bit vznikl ve Velké Británii v rámci iniciativy „Make it Digital“ soustředěné na rozšíření programování mezi mladší žáky. Projekt se měl postarat o nedostatek pracovní síly v rámci IT tím, že připraví novou generaci zájemců o digitální technologie [4]. Micro:bit byl vymyšlen s jasnými zásadami [5]:

1. Mít nízkou vstupní bariéru – být cenově dostupný a intuitivní, aby s ním jednoduše mohl pracovat každý žák.
2. Být zábavný, aby lehce upoutal pozornost žáků.
3. Být lehký pro začátečníky, ale zároveň poskytovat velké množství pokročilejších možností a zároveň obsahovat vícero způsobů, jak těchto možností dosáhnout.
4. Ukázat možnou budoucnost výpočetních technologií.
5. Být využitelný i mimo IT, bylo důležité, aby byl aplikovatelný i v jiných předmětech.

V roce 2016 mohl každý žák ve věku 11 až 12 let získat zdarma micro:bit. 90 % žáků se shodlo, že jim to pomohlo ukázat, že programovat může kdokoli, a 45 % uvedlo, že by v budoucnu považovali IT za kariérní volbu oproti původním 36 %. 85 % učitelů si myslelo, že je IT zábavnější, a 75 % by je chtělo používat v budoucnu [6]. Později se micro:bit začal

půjčovat i zdarma v knihovnách. Během roku 2018 se micro:bit dostal do povědomí po celém světě a stal se globálně prodáváným.

V roce 2020 vyšla verze micro:bit v2. Byl zde přidán reproduktor a mikrofón společně s novými softwarovými bloky v Microsoft Makecode. Dále bylo přidáno dotykové tlačítko na logo. Stejně tak bylo umožněno používat piny jako dotyková tlačítka. Byla přidána možnost ho uvést do režimu spánku. Tato verze má zároveň novější procesor, který slibuje být schopný zvládnout i strojové učení a umělou inteligenci a větší RAM. Celkově je tato verze v mnoha ohledech lepší než verze předchozí [7]. Nejnovější verze micro:bitu je v2.2, která vyšla v roce 2022. Došlo zde ke změně čipu, který se staral o USB komunikaci. Důvodem k této změně byl celosvětový nedostatek čipů. Pro uživatele se v praxi nic nezměnilo a vše zůstalo identické verzi v2 [8].



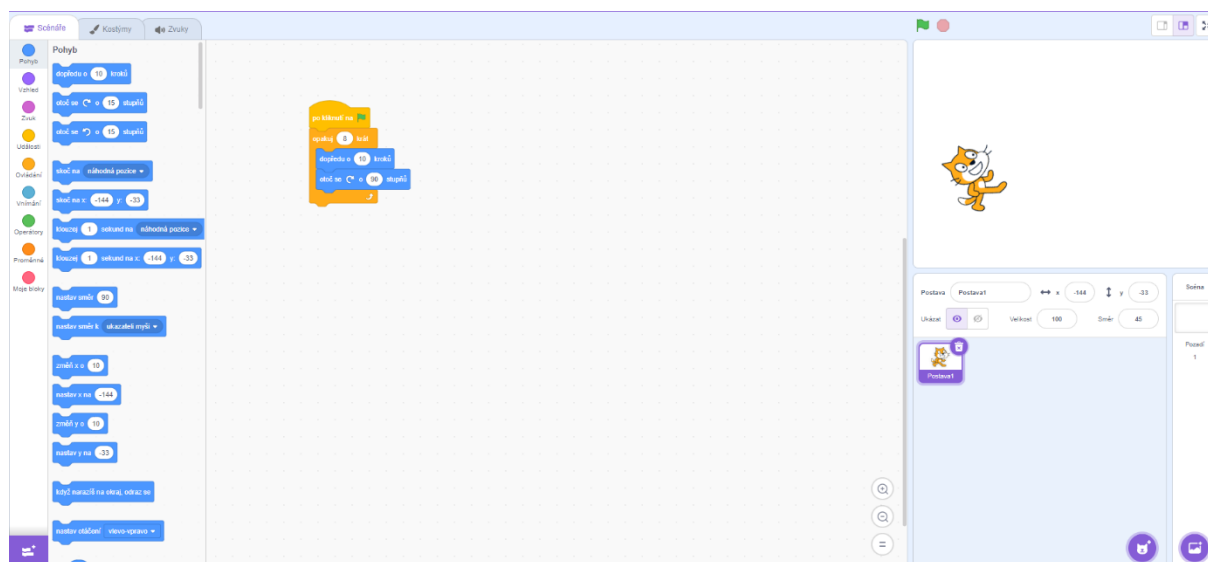
Obrázek 2: rozdíly mezi verzí 1.5 a 2 [7]

### 3. Bloková prostředí pro programování destičky micro:bit

Blokové programování narozdíl od textového nevyužívá psaného textu ale vizuálních bloků, které je možné lehce přetahovat a spojovat a díky tomu „stavět“ kód. Díky tomu by tento způsob programování mohl být vhodnější pro začátečníky a na učení základů. Jednoduše si mohou žáci otestovat své nápady, efektivně se naučí základy *for loop* a *what if* a pomůže jim to pochopit algoritmus v programování. Nevýhodou je, že v těchto programech se zřídka dají programovat složitější programy, tudíž eventuálně je nutné přejít na textové programování. Zároveň jsou programy často specializované a nejsou velmi univerzální.

#### 3.1. Scratch

Scratch je vizuální blokový programovací jazyk, který poskytuje jednoduché a pochopitelné prostředí pro začínající programátory. Je možné v něm dělat interaktivní projekty, animované příběhy, hry či simulace. Obsahuje mnoho addonů, kterými je možné rozšířit jeho možnosti. Žákům rozvíjí schopnost řešit problémy a díky jeho okamžité zpětné vazbě si může žák jednoduše ověřit svou logiku a způsob myšlení [9]. K používání nejsou potřeba předchozí znalosti, jazyk je intuitivní a díky využitému prostředí a ikonám i přístupný dětem. Pro spojení s micro:bitem je nutné nainstalovat aplikaci Scratch Link a přidat si rozšíření pro micro:bit. Samotné rozšíření neobsahuje mnoho nových bloků.



Obrázek 3: rozhraní Scratche

Rozhraní se skládá z blokové části (levá), programové části (uprostřed) a scény (vpravo). Uživatel vybírá bloky z levé strany, vkládá je do střední a na scéně po spuštění vidí své výsledky. Do scény lze přidávat další postavy či pozadí, kdy každé postavě lze nastavit vlastní kód. Tímto způsobem je možné vytvořit dialogy či komplexnější interakce. Výhodou je,

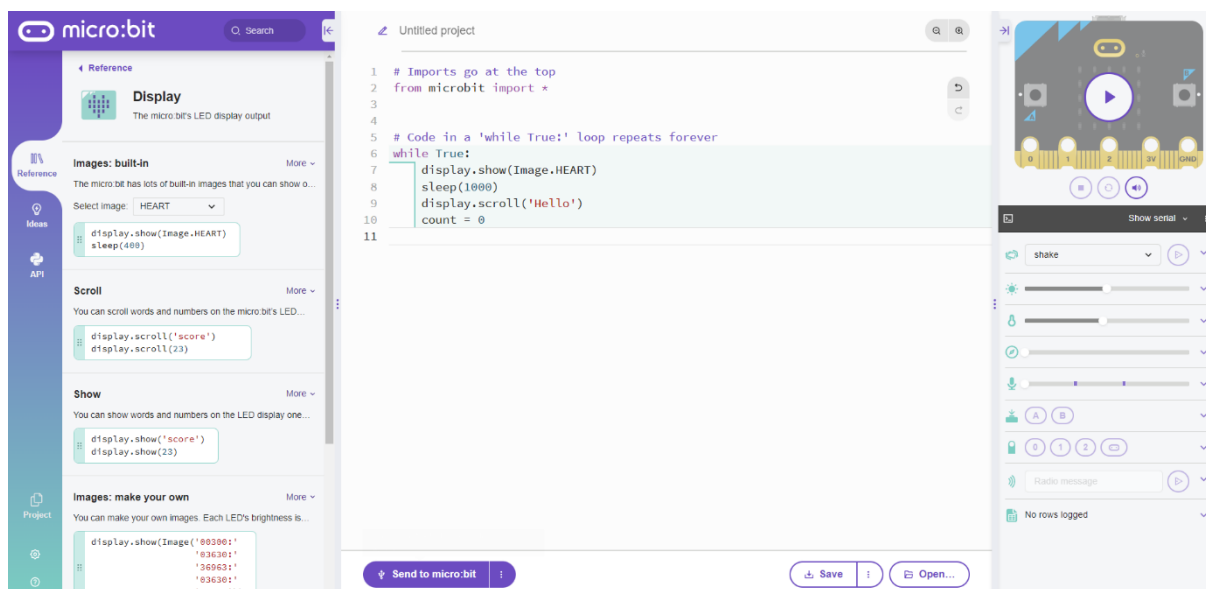
že není potřeba žádný hardware, takže si to každý může spustit doma. Nevýhodou pro naše využití je nepřítomnost softwarového simulátoru micro:bitu [1].

### 3.2. Python editor

Python editor je kombinace mezi blokovým programováním a textovým programováním. Přesouváním bloků se zapisuje kód, který v případě potřeby můžeme i ručně psát. Na rozdíl od Scratche a Makecode se do něj nedají instalovat další rozšíření. Má však záložku api, která obsahuje kódové bloky pro možné přidané periferie. Do micro:bitu jde následně kód nahrát přímo z webového prostředí.

Pokud budeme chtít psát kód ručně, obsahuje chytrý našeptávač, který je připravený pro příkazy do micro:bitu. Obsahuje dokumentaci, u každého bloku kódu můžeme rozkliknout „more“, což nám ukáže bližší popis a rady, jak se dá případný blok použít. Umožňuje vytvořit složitější programy než makecode, zároveň má jednodušší zápis podmínek a lze s ním provádět podrobnější práci s měřenými daty.

Také obsahuje softwarový simulátor micro:bitu, takže si napsaný kód můžeme vyzkoušet, než ho nahrajeme. Zatím bohužel nemá český překlad, zároveň s ním lze jednodušeji udělat chyby, proto je lepší pro pokročilejší žáky [2].



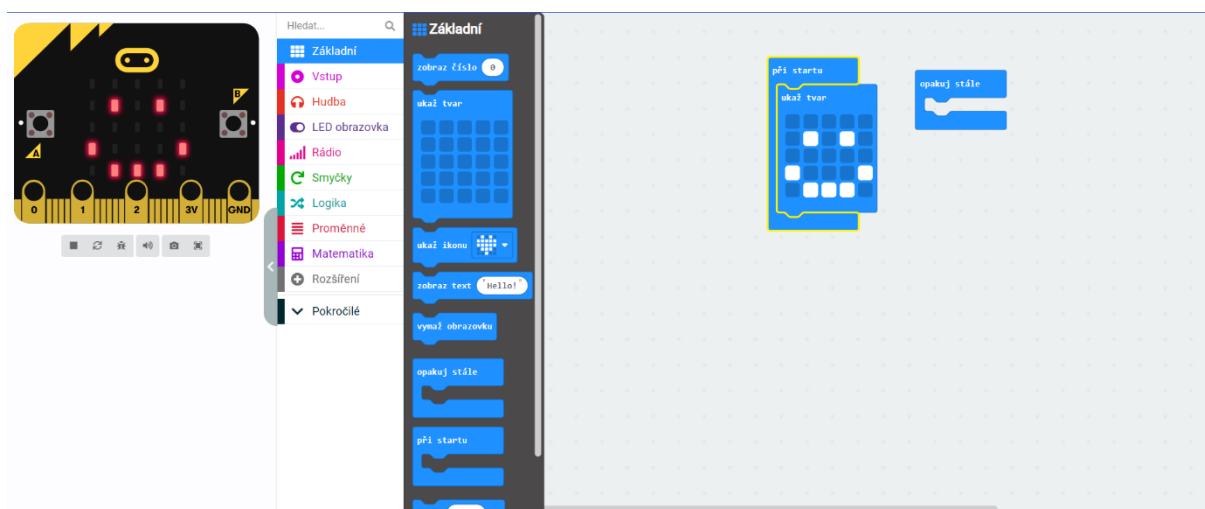
Obrázek 4: rozhraní Python editor

Rozhraní je intuitivní, napravo vidíme simulaci micro:bitu, kde můžeme nastavit hodnoty senzorů, stisknout tlačítka či ovlivnit gyroskop. Zároveň si můžeme zobrazit konzoli pro výpis dat. Uprostřed je samotný kód. Vlevo je knihovna možných bloků na přidání, jako první je zde například část kódu, která nám zobrazí srdce.



### 3.3. Microsoft Makecode

Microsoft Makecode, stejně jako Scratch a Vittascience, je webové blokové prostředí. Umožňuje nám vytvářet kód pro náš micro:bit a zároveň ho i do micro:bitu sám nahrát. Veškeré jeho bloky jsou přizpůsobené práci s micro:bitem a stejně jako u Scratche do něj lze přidávat další rozšíření, které ale často vyžadují další nakoupené komponenty. Je to také v češtině, tudíž zde není problémem jazyková bariéra. Bloky nelze sestavit syntakticky špatně, proto se žáci nemusejí soustředit na hledání středníků a závorek a musí se pouze soustředit na infromatické myšlení [2]. Díky tomu je to pro naše účely nejlepší volba.

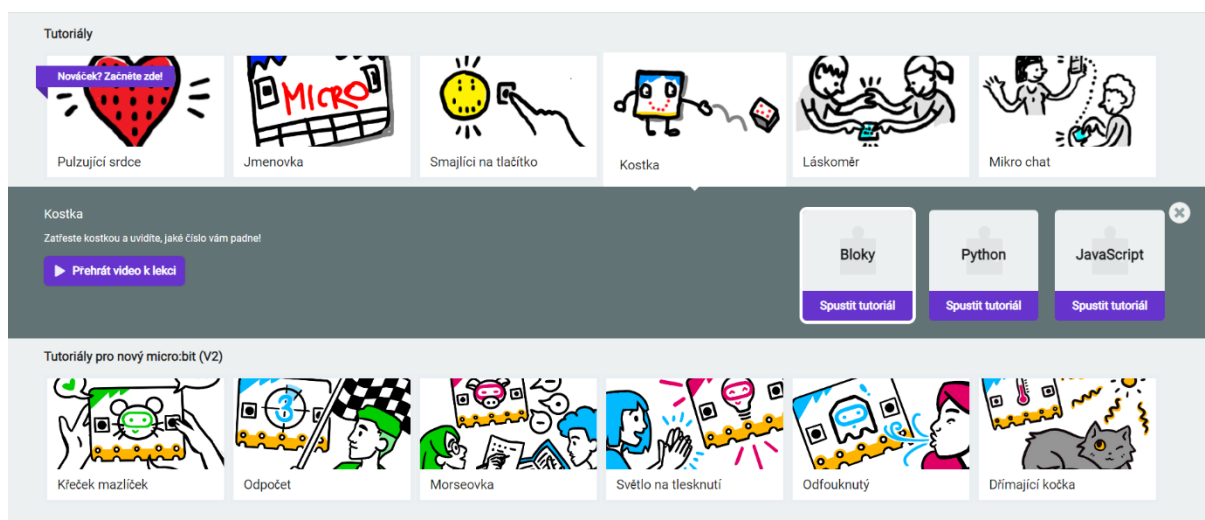


Obrázek 5: rozhraní Microsoft Makecode

Rozhraní v Makecode je velice intuitivní. Napravo je plocha, do které vkládáme bloky s kódem. Uprostřed máme panel s kategoriemi bloků, kde po rozkliknutí dostaneme celou nabídku. Vlevo vidíme simulaci micro:bitu, díky které nemusíme na každou změnu v kódu kód znovu nahrávat přímo do fyzického micro:bitu, ale vidíme zde v reálném čase, jak se bude chovat, do micro:bitu tedy můžeme nahrát až výsledný kód. V levém dolním rohu je tlačítko stáhnout, které stáhne náš kód do micro:bitu, případně do počítače, pokud micro:bit není připojen. Po stisknutí se zároveň objeví okénko s textem a animací, které nám v případě problémů poradí, jak správně micro:bit nastavit.

Výhoda tohoto rozhraní je možnost kód jedním stisknutím převést do Pythonu či Javascriptu a tím mohou žáci vidět svůj kód i v textové verzi. Tato funkce usnadňuje žákům přejít z blokového programování do textového programování a jsou díky ní vidět jasněji funkce jednotlivých příkazů a zároveň i syntax kódu.

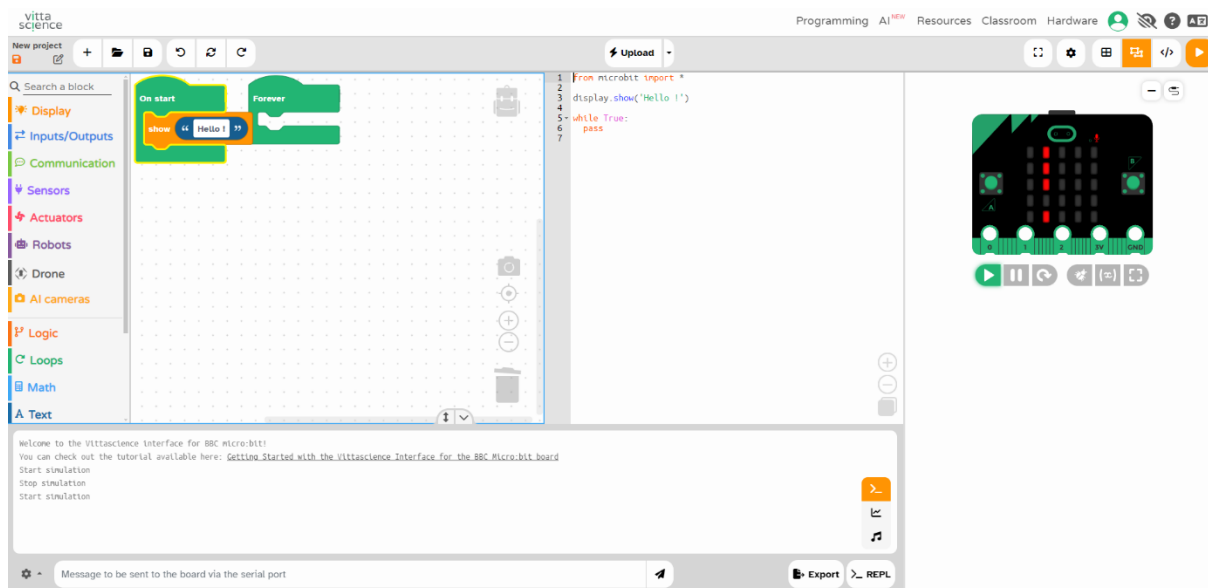
Samotná stránka Microsoft Makecode obsahuje velké množství tutoriálů a projektů na vyzkoušení. Tutoriály jsou sepsané krok po kroku společně s vysvětlením. Většina tutoriálů je pro blokové prostředí, ale některé nabízí i možnost tutoriálu v Pythonu nebo Javascriptu. Témata jsou různorodá, jsou zde tutoriály pro úplně začátečníky, například tvorba hodu kostkou či animace srdce. Jsou zde tutoriály jak pro starší verzi micro:bitu tak i pro tu novější v2.



Obrázek 6: možné tutoriály u Makecode Microbit

### 3.4. Vittascience

Vittascience je na první pohled velice podobný prostředí Microsoft Makecode. Opět se přesouváním bloků přidává kód. Vittascience narozdíl od Microsoft Makecode poskytuje bez rozšíření pokročilejší bloky a bloky pro práci s AI kamerou, drony a roboty. Pro mé účely nebyly tyto bloky potřebné. Nově Vittascience umožňuje i trénovat si vlastní AI model, který se dá následně použít společně například s kamerou nebo mikrofonom. Bohužel celá stránka je v angličtině, v některých částech jsem dokonce narazil na části, které byly pouze ve francouzštině. Na rozdíl od Microsoft Makecode neobsahuje předpřipravené projekty, které mohou žákům pomoci do začátku [10].



Obrázek 7: rozhraní Vittascience

Rozhraní je velmi podobné předchozím rozhráním. V levé části jsou kategorie bloků, které po otevření obsahují specifické bloky. Prostřední část je rozdělena na část s bloky a část s Pythonem, kdy do každé můžeme libovolně vkládat bloky či kód a automaticky se tím vyplňuje druhá. Je možné si zvolit mít otevřené pouze jedno takové okno. V pravé části máme simulaci samotného Micro:bitu, která je opět interaktivní a můžeme v ní používat tlačítka. V dolní části můžeme mít konzoli, která nám dává výpis o simulaci. Osobně mi přijde toto UI méně intuitivní a vzhledově více zastaralé než UI u Microsoft Makecode, zároveň rozdělení samotných bloků může být pro nějaké žáky moc komplikované a složité k orientaci.

### 3.5. Shrnutí

Po zvážení všech těchto rozhraní jsem došel k názoru, že pro mé účely bude nejvhodnější prostředí Microsoft Makecode. Scratch neobsahuje simulaci micro:bitu a v porovnání s Microsoft Makecode toho neobsahuje mnoho navíc. Vittascience je bohužel pouze v angličtině, poměrně nepřehledné a bloky navíc, které obsahuje, nejsou pro mé účely použitelné vzhledem k tomu, že ani nevlastním potřebné periferie. Python editor mě osobně zaujmul, poskytuje možnost se učit kód a zároveň se žáci nemusí učit úplně celou syntax nazpaměť. Zároveň obsahuje poměrně detailní dokumentaci a velké množství API v případě potřeby. V případě, že bych chtěl pracovat s micro:bitem pro osobní použití, bych určitě použil toto rozhraní. Bohužel je též v angličtině a zároveň může být pro žáky základní školy složitější než samotné bloky, tudíž jsem se rozhodl zvolit bezpečnější možnost a vybral si Microsoft Makecode. Obsahuje vše, co potřebuji, má velké množství rozšíření, obsahuje mnoho

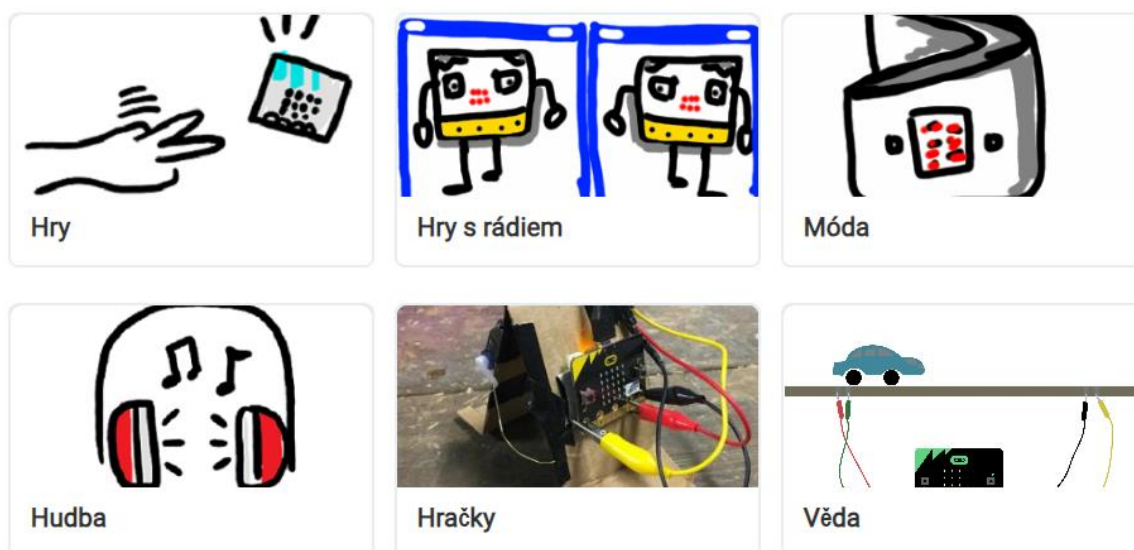
předpřipravených projektů a zároveň je vše v češtině. Má intuitivní UI a podporuje obě verze micro:bitu. Řekl bych, že pro potřeby žáků základní školy je toto rozhraní ideální, a proto jsem si ho nakonec zvolil.

## 4. Úlohy projektové výuky dostupné na internetu

Pro micro:bit existuje velké množství dostupných zdrojů s cvičeními a projekty, které mohou žáci ale i učitelé či kdokoliv jiný využít pro procvičování. Vybral jsem zdroje, které mi přišly dobře vytvořené a připravené, které splňovali zásady projektové výuky a kterými jsem se zároveň inspiroval při tvorbě vlastních úloh.

### 4.1. Makecode Microbit

Stránka Makecode Microbit kromě tutoriálů obsahuje i stránku obsahující mnoho projektů. Tyto projekty jsou rozděleny do vícero kategorií, jsou zde vědecké projekty zaměřené na měření pomocí senzorů, například pro měření rychlosti objektu nebo vlhkosti u rostlin. Dále jsou zde projekty zaměřené na hry, které si žáci mohou společně vyzkoušet při výuce. Zajímavá kategorie jsou například také nástroje, kde žáci vytvářejí kompas nebo vodováhu. Ne všechny projekty na této stránce splňují zásady projektové výuky, některé jsou pouze tutoriály. Další nevýhodou této stránky je fakt, že po rozkliknutí projektu je jich většina anglicky, což může žáky mást.

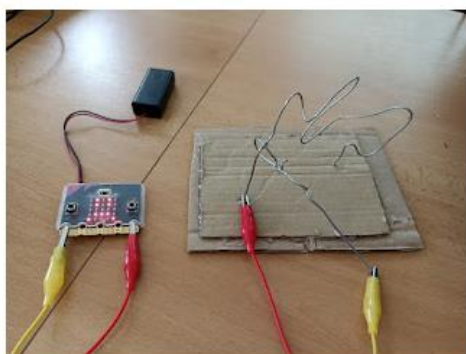


Obrázek 8: kategorie projektů u Microsoft Makecode [17]

## 4.2. Microbiti

Česká stránka Microbiti obsahuje menší množství projektů, zato ale kompletně v češtině. Obsahuje například vytvoření vlastní kytary nebo vytvoření voleb. Každý projekt obsahuje ukázkové řešení, které je obohaceno názornými fotkami a nápady na vylepšení [2].

### Test šikovnosti



Klasická hra z pevného drátu a drátěného očka, snažíme se bez doteku projet z jednoho konce na druhý.

Na základní verzi stačí baterka a žárovka (ledka), vylepšení s vypínačem.

Microbit umožňuje například počítat doteky, odstartovat hru, vydávat varovné zvuky a podobně.

Obrázek 9: test šikovnosti na stránce Microbiti [2]

## 4.3. Učíme s roboty

Další česká stránka jménem „Učíme s roboty“ obsahuje nejen projekty pro micro:bit, ale i projekty pro 3D tisk nebo Ozobota. Úlohy jsou pěkně popsány, autor popisuje své zkušenosti a svůj způsob provedení, zároveň vše vysvětluje a dává detailní návod, jak úlohy replikovat. Úlohy jsou pojaty stylem hry, ač nejsou moc praktické, tak pro žáky jsou určitě zajímavé, a dávají jim možnost se mnoho nového naučit a zároveň se pobavit. V jedné úloze dokonce žáci chodí po škole a hledají své bloky kódu. Často je zde zapojena i 3D tiskárna. Využívá i umělecké činnosti, kdy se micro:bit lepí ke kartónu, na který následně žáci mohou nakreslit nějakou postavičku, a celá aktivita je pro ně tak zajímavější. Úlohy jsou určeny pro mladší žáky, ale svým tématem a komplexitou by se dali použít i se staršími. Autor samotný píše, že není informatik, a popisuje zde své myšlenky a názory, proto se tato stránka hodí i pro učitele, kteří ač neučí informatiku, tak by chtěli například micro:bit zakomponovat do svých hodin. Jediná výtku, kterou mám pro tuto stránku je, že zde úloh není mnoho, ale pro inspiraci nejen pro micro:bit, ale i pro vedení samotné projektové výuky je stránka užitečná.



## Postřehová hra s Microbitem

ucimesroboty / 📅 1. 6. 2022 / 🖱️ BBC micro:bit

Nedávno jsme během projektových dnů vyzkoušeli další projekt s microbitem. Tentokrát se jednalo o hru postavenou na rychlosti reakcí, originální návod je zde:

<https://microbit.org/projects/make-it-code-it/reaction-game/>

V zásadě se používá princip tlačítka a uzavření elektrického obvodu, který jsme využívali už v [minulém projektu](#).

Varianta provedení je několik, na obrázku níže vidíte variantu se dvěma samostatnými tlačítky pro hráče A (pin 1, červený kabel) a hráče B (pin 2, žlutý kabel). Oba bílé kabely jsou zapojené do GND a tím uzavírají obvod.

Pokud se obvod uzavře (alobal na horní straně kartonu se dotkne alobalu na dolní straně), microbit ukáže buď A nebo B, v závislosti na tom, který obvod se uzavřel dříve – který hráč tlačítko zmáčkne rychleji.



Obrázek 10: příklad ze stránky Učíme s roboty [18]

#### 4.4. Oficiální stránka Microbit

Stránka Microbit též obsahuje vlastní projekty řazené dle obtížnosti. Způsob, kterým jsou projekty sepsané a vymyšlené, je dle mě nejlepší ze tří zde zmíněných možností. Každý projekt obsahuje úvodní popis, co bude potřeba, koncept fungování a některé mají i názorná videa. Každý projekt pak zároveň obsahuje možné řešení a nápady na vylepšení. Všechny projekty se dají řadit podle obtížnosti, tématu a použitých periférií. Celá stránka je v angličtině, vyjma ukázkových řešení, která používají přeložený Makecode. Velké množství projektů a řazení umožňuje mnoho možností pro inspiraci do projektové výuky [4].

The screenshot displays the 'Make it: code it' website interface. On the left, there is a search bar and filter options for programming languages (MakeCode (31), Python (22), Scratch (0)) and subjects/topics (Computational thinking, Computer systems, Digital arts, Networks, Programming, Safety & security, Design & technology, Geography). The main content area shows a grid of project cards, each with a title, a brief description, and a difficulty level indicator (Beginner, Intermediate, Advanced). The projects shown are: 'Sensitive step counter' (Intermediate), 'Compass North' (Intermediate), 'Max-min thermometer' (Intermediate), 'Make some noise' (Intermediate), 'Frère Jacques loops' (Intermediate), and 'Group teleporting duck' (Intermediate). On the right side, there is a list of 'Other projects' including 'Awesome micro:bit', 'Code Club', 'hackster.io', 'micro:bit in Wonderland', 'Microsoft MakeCode', and 'Raspberry Pi'.

Obrázek 11: projekty na stránce Microbit [4]



- Step 1: Make It
  - What is it?
  - What you'll learn
  - How it works
  - What you need
- Step 2: Code It
- Step 3: Improve It

## What is it?

A step-counter you can make more accurate by tailoring it to your own walking style.



## What you'll learn

- How the micro:bit can collect numerical acceleration data
- How to apply thresholds to sensor data to trigger events like increasing a step counter

## How it works

- The [Step counter](#) and [Low energy step counter](#) projects use the 'shake' gesture to count steps. The 'shake' gesture uses several [accelerometer](#) sensor readings to decide if the micro:bit has been shaken.
- You may find that the 'shake' gesture isn't triggered every time you take a step, or that it's triggered too easily, leading to inaccurate counting of steps.
- To make a more accurate step counter, instead of using the 'shake' gesture, this program uses numerical data from the accelerometer to decide whether you've taken a step and, if you have, increase the `steps` variable by 1.
- If the acceleration is greater than (>) 1500, the `steps` variable is increased by one and show the step count on the LED display output. 1500 is the `threshold` - the point at

Obrázek 12: popis projektu na Microbit [4]

## 5. Projektová výuka

Projektové vyučování je výuka založená na projektové metodě. Za projekt se považuje komplexní pracovní úkol, při kterém žáci samostatně řeší určitý problém. Charakteristickým výstupem je výrobek, tj. praktické řešení problému [11]. Žákům dává možnost vyjádřit zde svou kreativitu, v případě, že každý žák vytvořil na konci identický produkt, nebyla projektová výuka provedena správně [12].

Projektové vyučování usiluje o doplnění nedostatků tradiční školy jako například izolovanost učiva, odtrženost od života, mechanické učení a nízká motivace žáků. Pokud je provedena kvalitně a promyšleně, *stává se hodnotným doplněním a zkvalitněním výuky*. [13]

Díky tomu, že žáci řeší problémy, které jsou komplexní a které odpovídají více skutečnému světu, získávají žáci poznatky a prostor pro rozvoj kompetencí a kapacit. *Zdrojem poznání v tomto případě není pouze učitel či učebnice, ale „život“ sám* [14]. Učitel se dostává do role rádce, pozorovatele, pomocníka, moderátora nebo konzultanta. Na rozdíl od klasické výuky nemá klasickou roli učitele, ale nechává prostor žákům ke komunikaci mezi sebou. Žáci mají možnost pracovat ve skupinách a rozvíjejí si tak komunikační a sociální dovednosti. [15]. Je dobré, aby samotné žáky daný projekt bavil, pokud žáci dělají to, co je baví, tak lépe pracují, zapamatují si celou zkušenost, ucelí společné vztahy a jako bonus je jednodušší s nimi pracovat [12].

### 5.1. Postup při projektovém vyučování

Pro správné provedení projektové výuky je důležité si ujasnit, jaké kroky musí učitel provést [13] [15] [16]:

1. Fáze přípravy – Toto je fáze, ve které učitel určí cíl a úkoly. Je nejdůležitější, záleží zde na správném zvolení tématu či situace, které žáci budou řešit – je vhodné zvolit takové, které budou žákům nějakým způsobem blízké. Úkol by měl být žákům blízký a je vhodné, pokud nabízí nějaké spojení s jiným předmětem. Lze ocenit i iniciativu a návrhy samotných žáků.
2. Fáze plánování – Učitel se žáky naplánuje způsob, jakým se bude problém řešit. Vyhledají si dostupné informace, určí se forma výsledku. Zde je důležitá diskuse mezi žáky a učitelem, aby každý měl představu finálního produktu. Celá tato fáze je společný „brainstorming“ mezi učitelem a žáky. Pro každého žáka musí být na konci jednoznačné, co bude dělat.

3. Fáze provedení – Toto je fáze, kde žáci provádí zadanou činnost. Každý by měl vědět, co má dělat, žáci pracují s dostupnými zdroji a pracují na splnění zadání. Je vhodné, aby žáci mohli diskutovat mezi sebou a prezentovat své výsledky před ostatními, aby cítili větší zodpovědnost za svůj projekt.
4. Fáze hodnocení – Zde žáci odprezentují svůj projekt, ukážou ho učitelům a ostatním. Je důležité, aby zde učitel zhodnotil především průběh celého projektu, spolupráci žáků a jejich aktivitu, a ne pouze výsledek samotný. Často je dobré tedy použít slovní hodnocení i sebehodnocení a vzájemné hodnocení mezi žáky. Je důležité brát v potaz i ostatní faktory jako například úroveň inteligence žáků či jejich zkušenosti s daným tématem a neodrazovat žáky nevhodnými poznámkami o jejich neschopnosti, pokud se jim projekt nedaří podle jejich představ. [13] [15]

Následuje reflexe – zde se učitel i žáci ohlédnou za celým projektem a shrnou si, co nového se naučili a jaký přínos projekt měl. Zpětná vazba může proběhnout formou pouhé diskuse, ale i formou písemnou, například esejí či dotazníkem. Zpětná vazba samozřejmě není důležitá pouze pro učitele, ale i pro žáky – je důležité, aby si uvědomili, jak pro ně projekt probíhal, jaká byla jejich role a co by příště mohli udělat lépe. Závěrečná reflexe a zpětná vazba jsou důležitou součástí celého projektu, pomáhají nám se vyvarovat chybám do budoucna [15].

## 5.2. Pozitiva a negativa projektové výuky

Výrazným pozitivem je ten fakt, že žáci řeší skutečné a komplexní problémy, díky tomu získávají nové znalosti přirozeně a souvisle, zároveň si je zapamatují jednodušeji a na delší dobu [16]. Zároveň si mohou rozvíjet své myšlenkové struktury a nadání, místo pouhého pasivního naslouchání. Díky propojení s ostatními předměty si můžou lépe dát naučené poznatky do souvislostí a vidět je lépe jako celek. Možnost práce na vlastním projektu dává žákům motivaci, obzvláště ve skupinových projektech, kde si žáci mohou zvolit co chtějí dělat. Každý si může najít něco svého a dělat to, v čem exceluje [13]. Dále mají žáci možnost pracovat v týmu, vzájemně si pomáhají, komunikují a učí se respektovat názor ostatních a zároveň vyjádřit ten svůj. Žákům toto pomůže rozvinout si své komunikační dovednosti a být schopni do budoucna lépe vést argument a diskusi [15] [16].

Projektová výuka má samozřejmě i své negativní stránky, primárně ze strany učitele. Velkým úskalím úspěšné projektové výuky je potřeba časově i psychicky náročné přípravy [16]. Pro správný průběh se zároveň předpokládá, že žáci budou mít potřebné znalosti pro splnění daných projektů. V případě, že toto není splněno, nemůže být projektová výuka zdaleka

tak efektivní, jak by mohla být. Zároveň učitel musí najít správnou bilanci mezi pouhým pozorováním a pomáháním žákům. Je důležité fungovat jako poradce, ale nechat žáky, aby si našli řešení pro své problémy sami. Při ponechání příliš velké volnosti ovšem může nastat situace, kdy žáci budou zkoušet vícero řešení metodou „pokus-omyl“ a během daného času zůstanou dlouho zaseknuti v jednom bodě, zároveň příliš velká volnost může způsobit neorganizovanost a zmatenost celé skupiny žáků. Je důležité v tomto tedy mít jistou rovnováhu a nenechat žáky se při projektu zbytečně dlouho „topit“. Projektová výuka už sama o sobě je náročná na časovou dotaci a v případě nesprávného provedení se může čas drasticky prodloužit [13] [15].

### **5.3. Shrnutí**

Projektová výuka dává možnost žákům si osvojit nové znalosti, či procvičit ty stálé způsobem, který má spojitost s reálným světem. Je to dobrý a zajímavý způsob, jak obohatit svou výuku a zároveň žáky motivovat k učení, ale i komunikaci mezi sebou. Není však jednoduché tento styl výuky provést správně a je důležité zvážit různá omezení a typ tématu, kterého by se výuka měla týkat. Není dobré jí tedy zakomponovat do výuky za každou cenu [11] [13].

## 6. Projekty

Praktická část se zabývá vytvořením projektů vhodných pro projektovou výuku s micro:bitem a odzkoušením samotných projektů ve výuce. Na základě zpětné vazby došlo následně k úpravě projektů a zadání.

V rámci projektové výuky jsem se rozhodl pro žáky vybrat téma chytrá domácnost. Nabízí se velké množství vychytávek, které si žáci mohou vymyslet a které jim zároveň přijdou užitečné. Žáci si zadání vyberou sami, ale i přesto jsem připravil několik ukázkových úloh, abych pomohl učitelům se inspirovat či inspirovat žáky. Navrhuji převážně takové úlohy, které používají externí periferie, aby si žáci vyzkoušeli nejen programování, ale i zapojování. U každé úlohy můžeme hodnotit to, jak žáci spolupracovali, jaké měli nápady, a i jak fungoval jejich kód. Je možné hodnotit i výsledný produkt, do kterého je micro:bit vložen, osobně bych se soustředil spíše na funkčnost než vzhled. Úlohy jsou koncipované pro micro:bit v1.5, který jsem měl já i škola, na které jsem je zkoušel, k dispozici. Výjimkou je poslední úloha, kterou jsem koncipoval pro micro:bit v2.0, jako ukázkou toho, čeho je novější verze schopná.

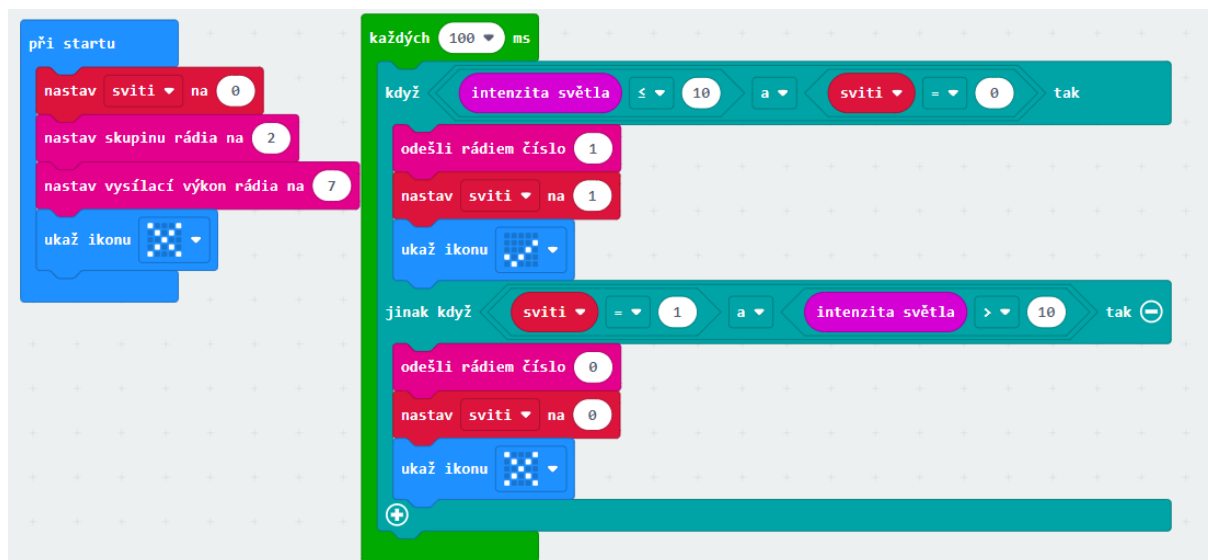
### 6.1. Automatická lampa

**Zadání:** Vytvořte automatickou lampu, která se bude zapínat a vypínat na základě detektoru světla. Použijte 2 micro:bity a diodu.

**Poznámka pro učitele:** Úlohou je automatická lampa do domácnosti – lampa, která detekuje světlo venku a v moment, kdy je tma, se rozsvítí. K sestavení této úlohy jsou potřeba 2 micro:bity a dioda. Jeden měří světlo a vysílá signál k zapnutí či vypnutí. Druhý je napojen na lampu, v našem případě na LED diodu, a po přijetí signálu ji rozsvítí. Poskytuji zde možné řešení, vše je však na žácích a mohou přijít s úplně jiným způsobem řešení.

**Cíl a hodnocení:** Žáci si ověří svou schopnost zapojit diodu a pracovat s piny. Dále žáci využijí, bloky rádia a senzor světla zabudovaný v micro:bitu. Je důležité pozorovat, zda žákům dojde, že při použití jednoho micro:bitu se lampa bude zapínat a vypínat dokola, protože bude vnímat svoje světlo, a proto potřebujeme alespoň 2 ke správnému fungování.

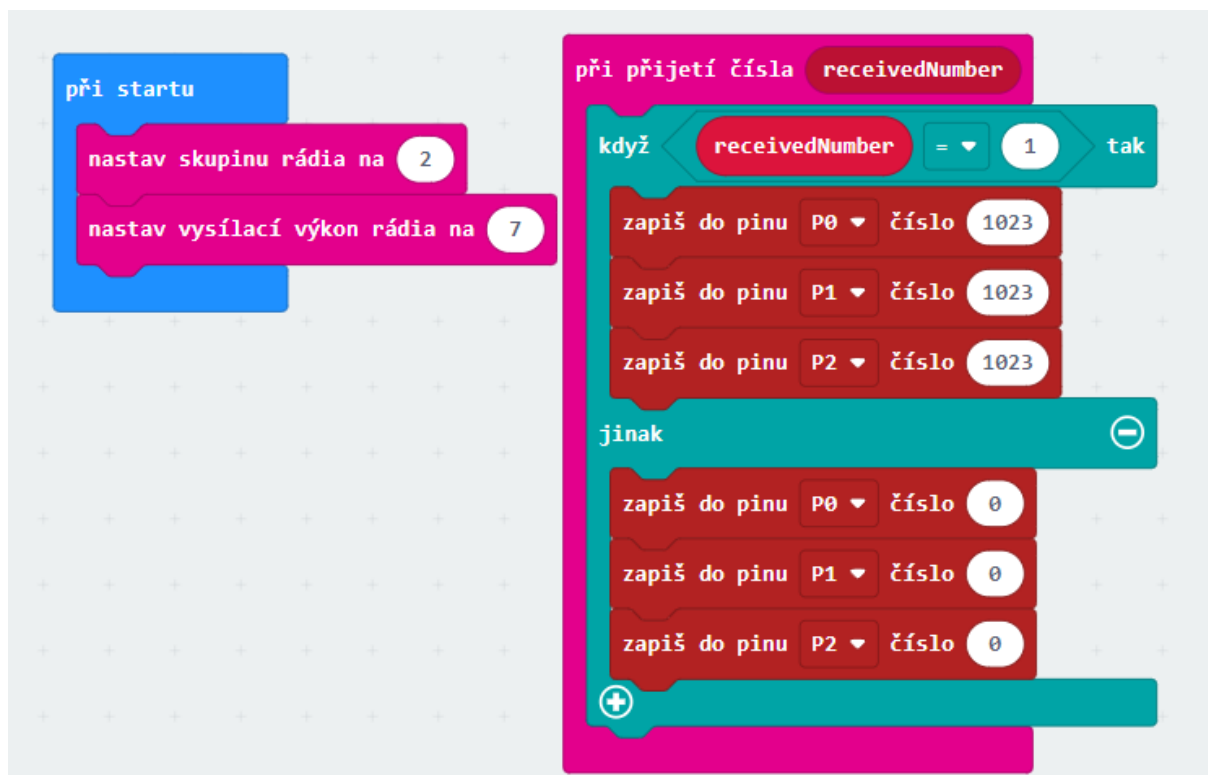
**Řešení pro senzor světla:** Jako první je nutné si sestavit samotný senzor světla. Využijeme vestavěného senzoru světla v micro:bitu. Dále je důležité chápat fungování bloků rádia.



Obrázek 13: kód pro senzor světla

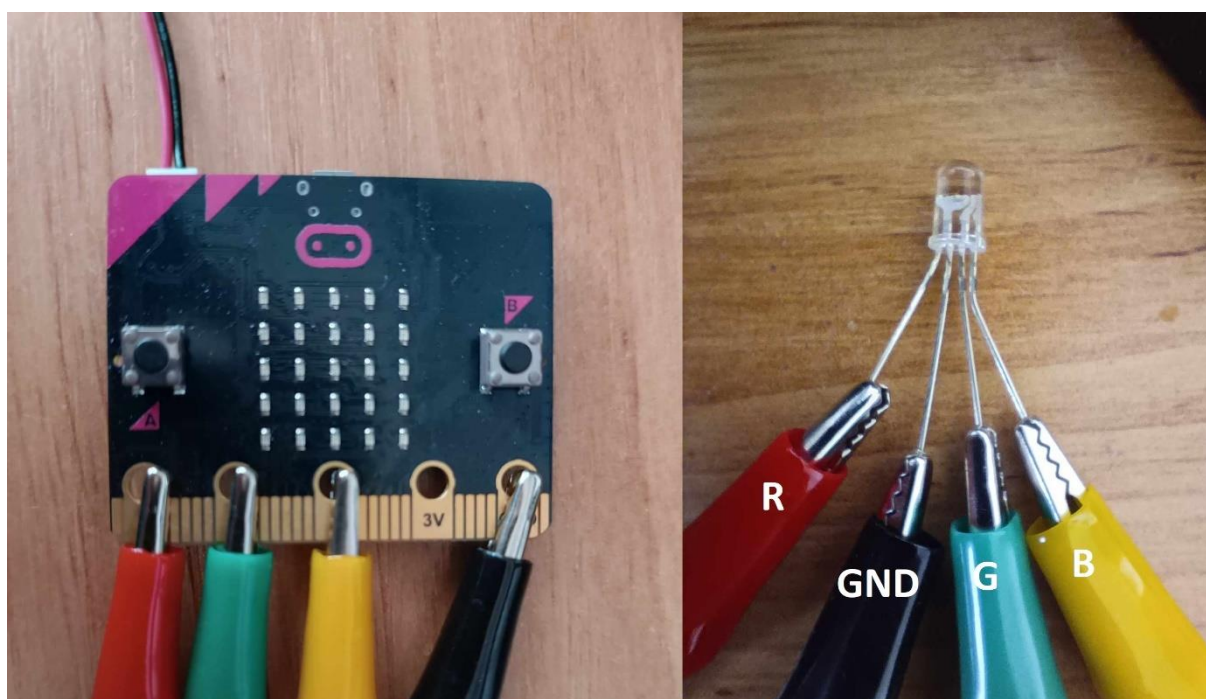
Zde je mnou navrhnutý kód pro senzor. Při startu se nastaví skupina rádia, zobrazí se ikona, že lampa je vypnutá, a nastaví se proměnná svítí. Tato proměnná nám určuje, zda je lampa zapnutá či ne. Následně se každých 100ms ověří, zda je v okolí dostatečné světlo, a v případě, že okolní světlo padne pod určitou hranici, vyšle se signál 1 a změní se ikona. Zároveň se ověřuje, zda již lampa není zapnutá, aby se neposílal signál zbytečně dokola.

**Řešení pro lampu:** U lampy připojíme LED diodu, můžeme vybrat normální či RGB. Pin si můžeme vybrat, jaký chceme, záleží na typu diody. Důležité je správně určit katodu a tu zapojit do GND.



Obrázek 14: kód pro lampu

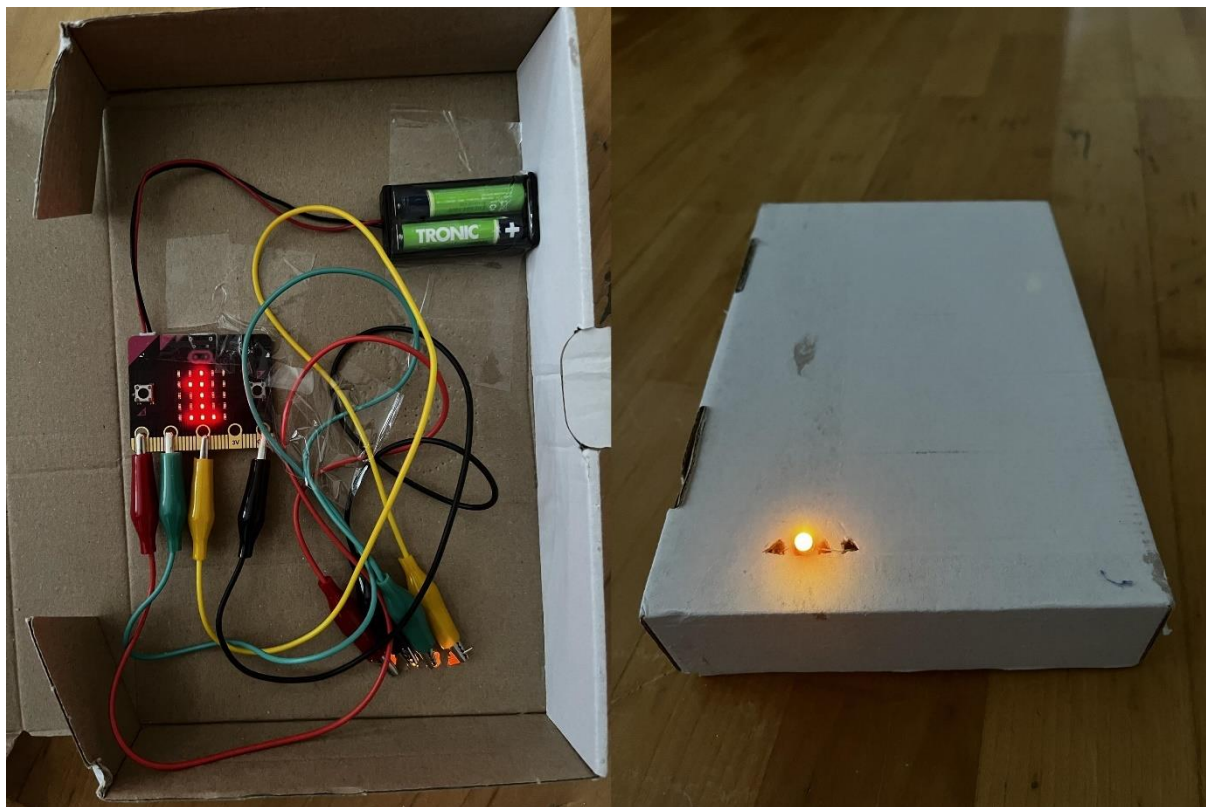
Toto je možný kód pro samotné zapínání. Používáme zde RGB diodu. Na začátku si tedy nastavíme správnou skupinu rádia a poté po přijetí čísla kontrolujeme, jaké číslo jsme dostali. V případě, že přišlo číslo 1, tak se všechny piny zapnou a dostaneme tím bílé světlo. V případě přijetí jakéhokoliv jiného čísla se všechny piny nastaví na hodnotu 0.



Obrázek 15: zapojení diody



**Zapojení a produkt:** U zapojení je důležité dát si především pozor na správné zapojení RGB diody. Nejdelší zakřivená část je anoda, ta přijde na micro:bitu do části GND. Zbytek jsou barvy, červená, zelená a modrá. Následně můžeme využít například kartónové krabice, do které celý micro:bit schováme a uděláme díry pro diodu. Výsledný produkt tedy může vypadat například takto:



Obrázek 16: produkt lampa



Pokud chceme, můžeme si k diodě přidělat i vlastní stínidlo pomocí papíru. Stačí si z papíru udělat válec, nalepit navrch kolečko z papíru a přilepit to k naší diodě. Může to ve výsledku vypadat například takto:



Obrázek 17: lampa se stínidlem

Samotný senzor musí být venku, není tedy dobré ho schovávat, může vypadat například takto:



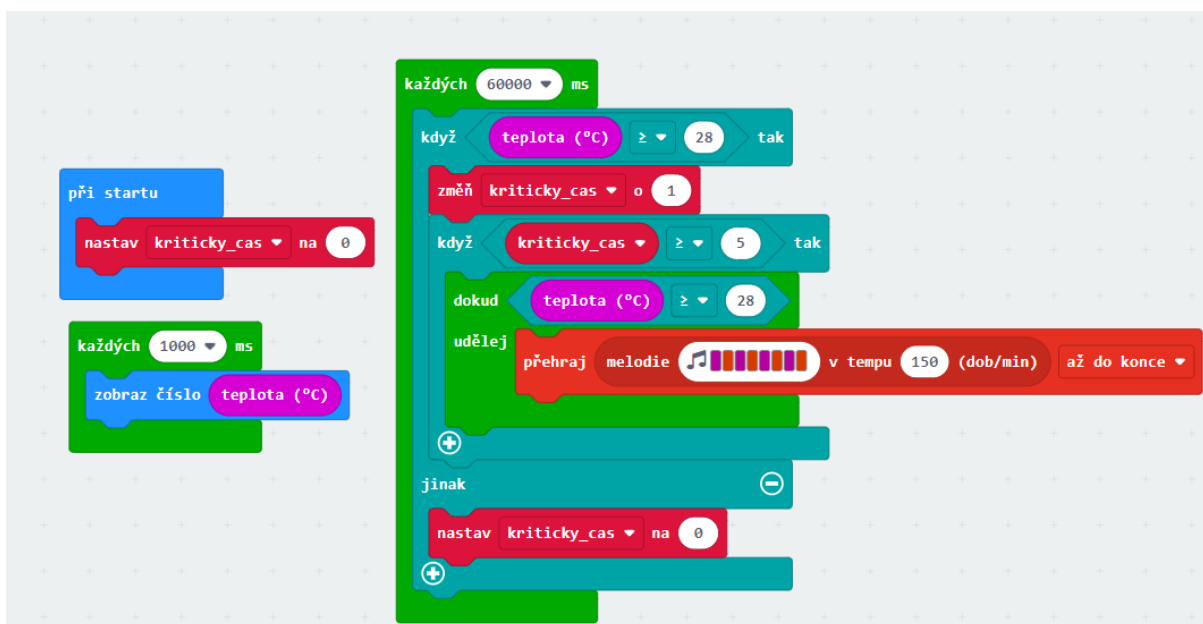
Obrázek 18: produkt senzor světla

## 6.2. Teploměr

**Zadání:** Vytvořte teploměr, který ukazuje teplotu a v případě přesažení limitu spustí varování. Využijte alespoň jednu periférii.

**Poznámka pro učitele:** Další možný projekt pro žáky je teploměr. Využijeme zabudovaného senzoru teploty v CPU. Žáci mohou buď využít dvou micro:bitů a měřit díky tomu teplotu venku. Toto má tu výhodu, že můžeme využít micro:bit z předchozí úlohy a nechat ho posílat do jiného micro:bitu i údaje o venkovní teplotě. Já se ovšem jako ukázkovou úlohu rozhodl připravit vnitřní teploměr, který hlídá teplotu, a v případě, že bude teplota moc dlouho vysoká, například pokud necháme moc dlouho zapnuté topení, nebo zase moc nízká, tak nás micro:bit hlasitým pípáním začne varovat. K sestavení této úlohy je potřeba bzučák či reproduktor.

**Cíl a hodnocení:** Cílem této úlohy pochopit, jak funguje vnitřní teploměr v CPU micro:bitu a využít ho pro měření teploty v okolí. Žáci zároveň pochopí zapojení bzučáku a jeho využití s bloky pro zvuk. Procvičí si své inženýrské myšlení využitím několika bloků „když“. Je dobré zde hodnotit vynalézavost žáků a jejich vlastní inovaci.



Obrázek 19: kód pro teploměr

**Řešení pro teploměr:** V ukázkovém řešení jsem si vytvořil proměnnou „kriticky\_cas“, která měří, jak dlouho byl přesáhnut limit teploty. Každou vteřinu zobrazuji aktuální teplotu. Každou minutu se kontroluje, zda teplota přesáhla nastavený limit. V případě že ano, zvýší se proměnná o 1. Pokud bude teplota přesáhnuta po dobu, kterou si nastavíme, v mém případě 5 minut, spustí se alarm, který nás varuje, že je na čase například vypnout topení. V případě, že teplota padne pod hranici, alarm se vypne a „kriticky\_cas“ se resetuje.

Žáci si zde mohou určit vlastní teploty, zvuk, který jim má hrát, nebo i dobu, po jakou musí být teplota přesažena. Je možné například přidat diody a alarm vytvořit i vizuálně.

**Zapojení a produkt:** U tohoto teploměru je důležité zapojit správně námi zvolenou periferii na varování, v tomto případě bzučák. Jeden pin zapojíme na námi zvolený pin na micro:bitu a druhý ke GND.

### 6.3. Alarm na dveře

**Zadání:** Vytvořte dveřní alarm. Použijte 1-2 micro:bity a alespoň jednu periferii.

**Poznámka pro učitele:** Další možná úloha pro žáky je alarm. Tato úloha má velké množství řešení a zároveň může využít velké množství komponentů jako například dioda, bzučák, tlačítka. V ukázkovém řešení využívám bzučák. Micro:bit bude připevněn na dveře a v případě, že zaznamená otevření, vyšle do druhého micro:bitu zprávu, aby spustil alarm. Otevření je možné kontrolovat vícero způsoby, já využívám kompas, ale je možné využít i možnost měření magnetického pole a zaregistrovat tak otevření pomocí magnetu.

**Cíle a hodnocení:** Tuto úlohu je možné řešit více způsoby. V každém případě žáci vymyslí způsob, jak detekovat, že se micro:bit nebo něco v okolí pohnulo. Pro kalibraci alarmu využijí pokročilejší funkci psaní do sériového portu. Zde můžeme hodnotit způsob jakým je toto detekováno. Žáci pochopí funkci rádia a vymyslí systém alarmu a jeho vypínání. Můžeme ocenit jejich inovaci a využití více periférií pro splnění tohoto úkolu.

**Kód pro senzor otevření:** Pro kontrolu otevření využívám kompasu. Ten je nutné nejdříve kalibrovat – při prvním spuštění se kalibrace automaticky spustí, otáčením micro:bitu se vyplní

```

při startu
  nastav skupinu rádia na 2
  nastav vysílací výkon rádia na 7
  nastav zapnuto na 0

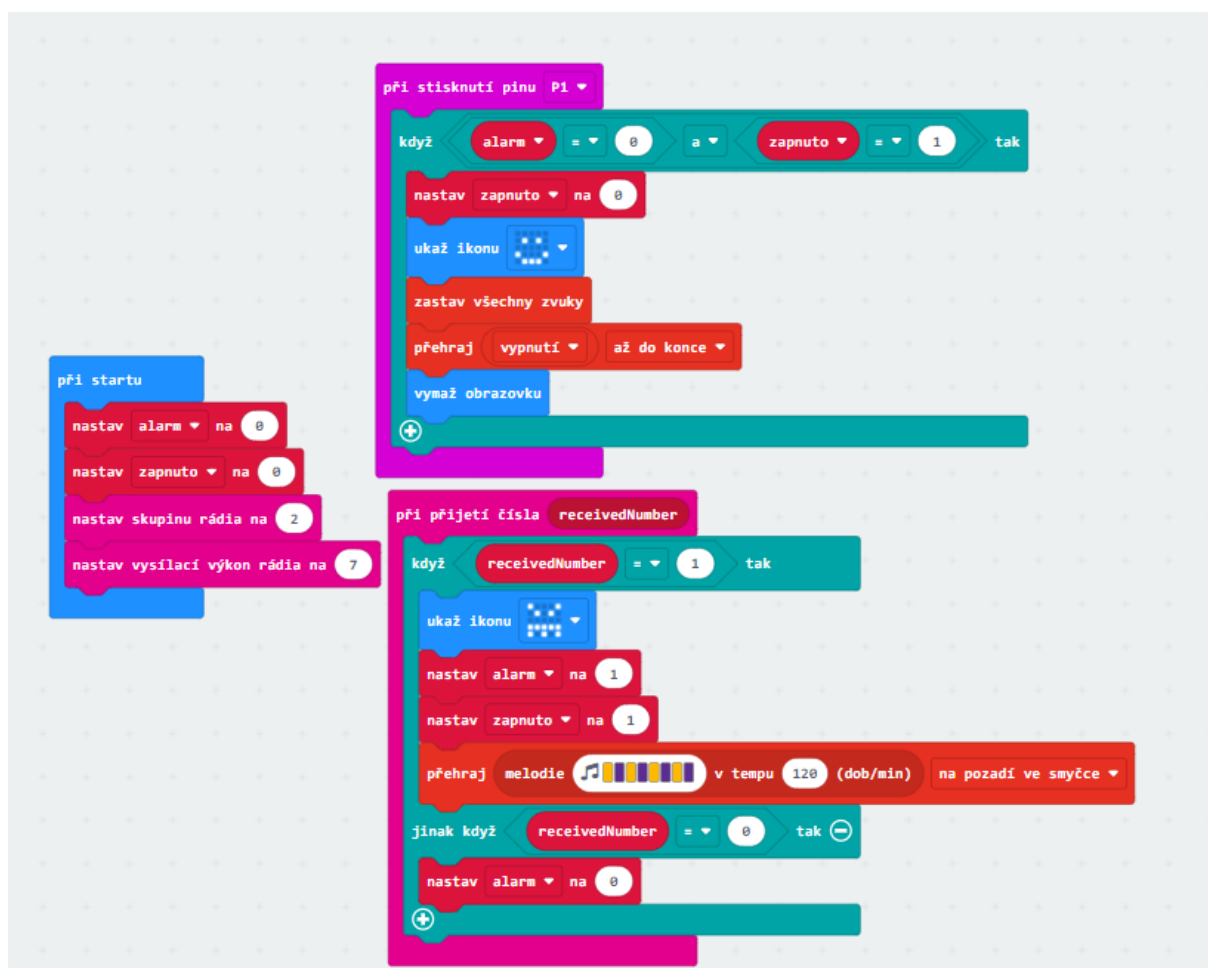
každých 100 ms
  zapiš číslo nasměrování kompasu (°) na sériový port
  zapiš řádek na sériový port
  když nasměrování kompasu (°) >= 100 a nasměrování kompasu (°) <= 160 tak
    když zapnuto = 1 tak
      nastav zapnuto na 0
      odešli rádiem číslo 0
      zobraz číslo 0
    jinak
      když zapnuto = 0 tak
        nastav zapnuto na 1
        odešli rádiem číslo 1
        zobraz číslo 1
  
```

Obrázek 20: kód pro senzor otevření

LEDky a kompas bude zkalibrován. Pro získání správných hodnot a nastavení, kdy jsou dveře považovány za zavřené, je vhodné využít pokročilé možnosti „sériová komunikace“, která nám do konzole zapisuje okamžitě naměřené hodnoty a která není zdržována vypisováním pomocí LED displeje. Po naměření potřebných hodnot a nastavení micro:bitu můžeme tuto funkci z kódu odstranit.

Toto je jedno z možných řešení pro senzor. Na začátku nastavím správnou skupinu a vytvořím si proměnnou „zapnuto“, která se stará o to, aby se neposílal stále stejný signál dokola. Každých 100ms micro:bit kontroluje, zda je jeho kompas mezi nastavenými hodnotami, které značí zavřené dveře. V případě, že se dostane mimo tyto hodnoty, vyšle do druhého micro:bitu signál alarmu. Pokud jsou zavřené na LED displeji se ukazuje 0, v opačném případě se ukazuje 1.

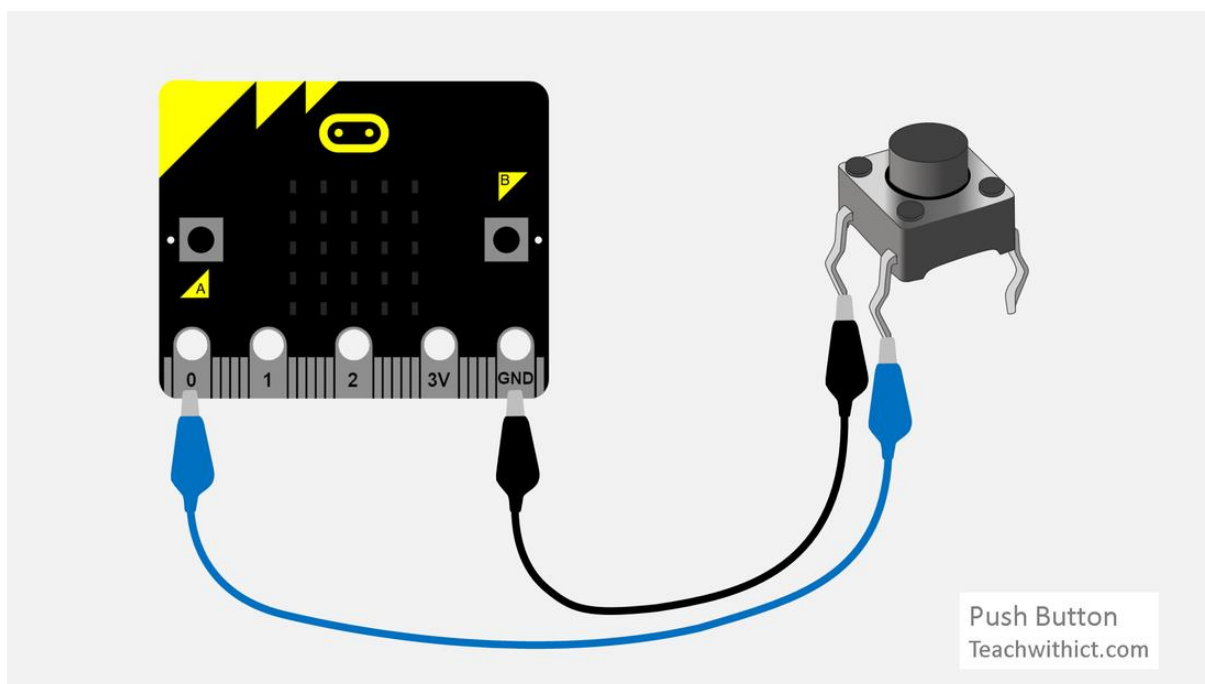
**Kód pro alarm:** Ke zvuku alarmu zde používám bzučák, je možné přidat například barevné diody. Po obdržení signálu se alarm spustí a je nutné ho manuálně vypnout. Žáci si zde mohou nastavit vlastní kombinaci tlačítek, pokud chtějí.



Obrázek 21: kód pro alarm

Mám zde 2 proměnné. Proměnná „alarm“ si ukládá poslední přijatý signál od senzoru. Proměnná „zapnuto“ kontroluje, zda je samotný alarm zapnutý. V případě, že přijde signál o spuštění alarmu, nastaví se tyto hodnoty na 1 a začne hrát melodie. Pokud chceme alarm vypnout je nutné, aby se dveře zavřely, tím se hodnota „alarm“ nastaví na 0 a poté musíme stisknout přidané tlačítko. Poté se alarm vypne a znovu čeká na spuštění.

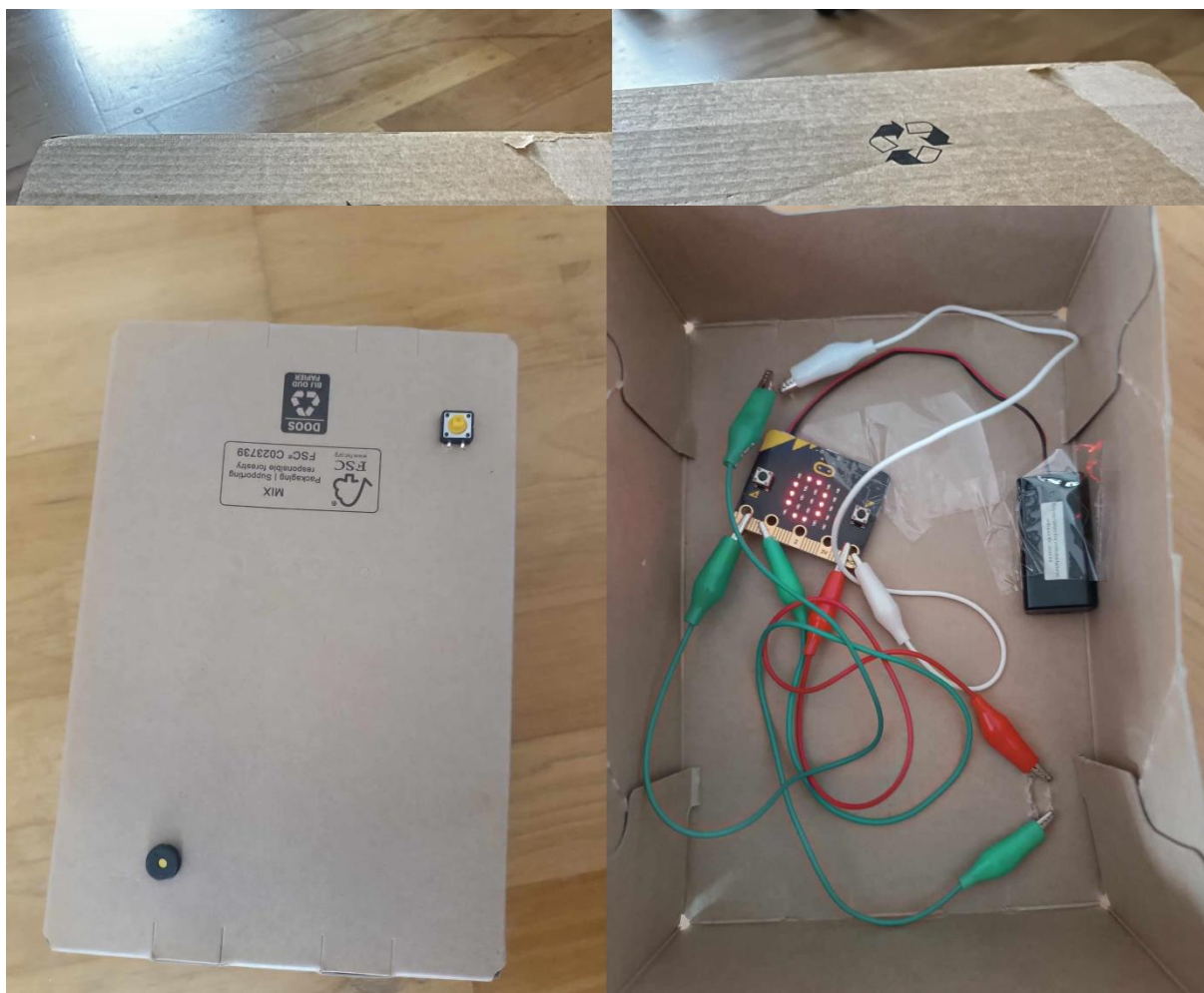
**Zapojení a produkt:** Na zapojení zde opět máme bzučák a tentokrát je k micro:bitu ještě přidané tlačítko. Bzučák zapojíme tak, že první pin zapojíme k pinu, který si zvolíme na



Obrázek 22: zapojení tlačítka [19]

micro:bitu. Druhý půjde do části GND. Pro zapojení tlačítka opět využijeme jednoho zvoleného pinu, a druhý půjde do GND. Je důležité je zapojit stejně jako na přiloženém schématu:

Následně tento produkt můžeme vzít a opět využít nějaké kartonové krabice. Pro vyzkoušení si můžeme z jedné krabice vyříznout provizorní dveře, na které micro:bit přilepíme.



Obrázek 24: produkt alarm

A pro samotný alarm si vzít druhou krabici, do které si uděláme díry pro tlačítko a bzučák a vše ostatní schováme dovnitř. Výsledný produkt může vypadat například takto:

#### 6.4. Zabezpečení proti krádeži

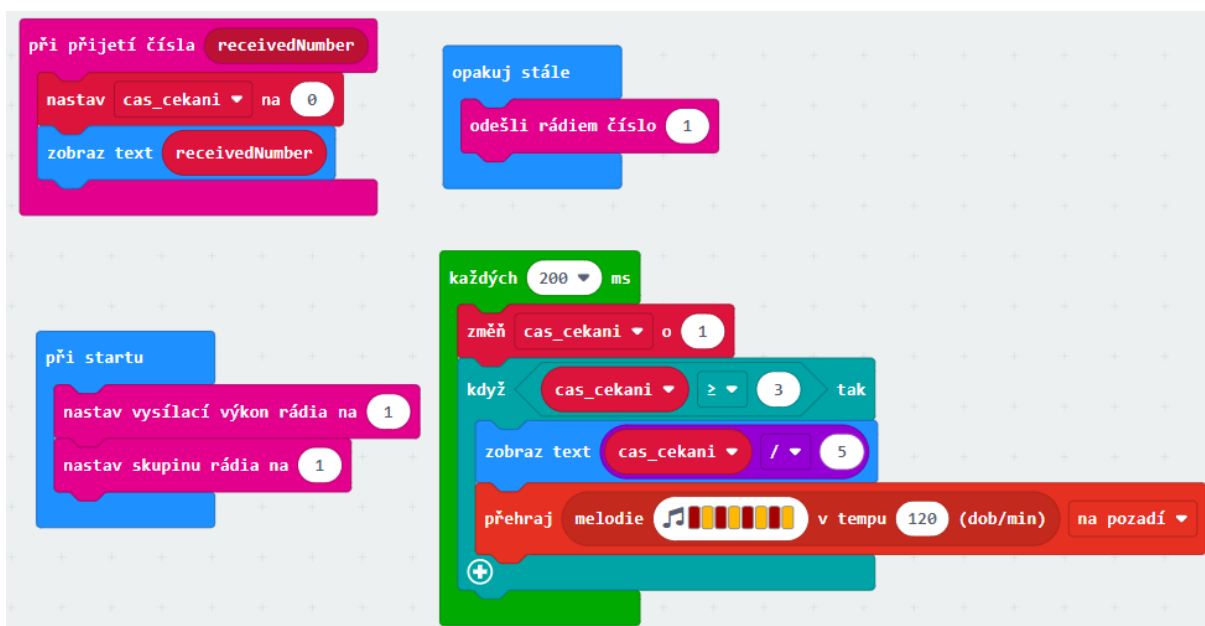
**Zadání:** Vytvořte zařízení, která spustí alarm v případě, že se od sebe vzdálí. Využijte 2 micro:bity a alespoň jednu periférii.

**Poznámka pro učitele:** Další úloha je zařízení, které zabezpečuje proti krádeži nebo ztrátě. Tato úloha vyžaduje 2 micro:bity a bzučák. Tyto 2 micro:bity se od sebe nesmí vzdálit na delší vzdálenost, v případě, že mezi sebou ztratí kontakt, spustí alarm, který se nevypne do doby, dokud se k sobě opět nepřiblíží. Celý princip je založen na tom, že si micro:bity navzájem



posílají signály a jakmile od sebe určený čas žádný signál neobdrží, spustí alarm. Na obou micro:bitech běží stejný kód, v teorii se dá takto propojit více zařízení.

**Cíle a hodnocení:** Žáci zde využijí pokročilejší funkce rádia, vymyslí způsob jakým hlídat vzdálenost mezi 2 micro:bity. Zároveň zde zapojí periférii, která dá o přílišném vzdálení vědět. Je zde důležité dát pozor na přesnost tohoto měření a to, zda micro:bity opět navážou spojení po přiblížení.



Obrázek 25: kód pro zabezpečení proti krádeži

**Řešení pro zabezpečení proti krádeži:** Toto je možný kód pro tuto úlohu. Je zde proměnná „cas\_cekani“, která si drží čas bez signálu. Pokaždé, kdy micro:bit přijme signál, se tato hodnota nastaví na 0. Každých 200ms se tato hodnota zvýší o 1 a micro:bit spustí bzučák, bude-li vyšší jak 3. Pokud tedy nedostane signál, aby časovač resetoval, spustí alarm, dokud se časovač znovu neresetuje. LED displej ukazuje čas, jak dlouho je mimo kontakt. Čas čekání a frekvenci si může každý nastavit podle sebe. Je důležité zmínit, že zde úmyslně nastavuji vysílací výkon rádia na 1, kdy každá hodnota znamená 10 metrů navíc. V případě, že bychom chtěli zvýšit vzdálenost, můžeme nastavit vysílací výkon až na 7, tedy 70 metrů.

**Zapojení a produkt:** Zde je opět důležité především zapojení námi zvolené periférie, která nás bude varovat při vzdálení micro:bitů. Já zde opět zvolil bzučák, kde opět zapojíme jeden pin do pinu na micro:bitu. Pro samotný produkt jsem zvolil pouhou destičku vyříznutou z kartónové krabice vzhledem k tomu, že je u tohoto zařízení důležitá kompaktnost.

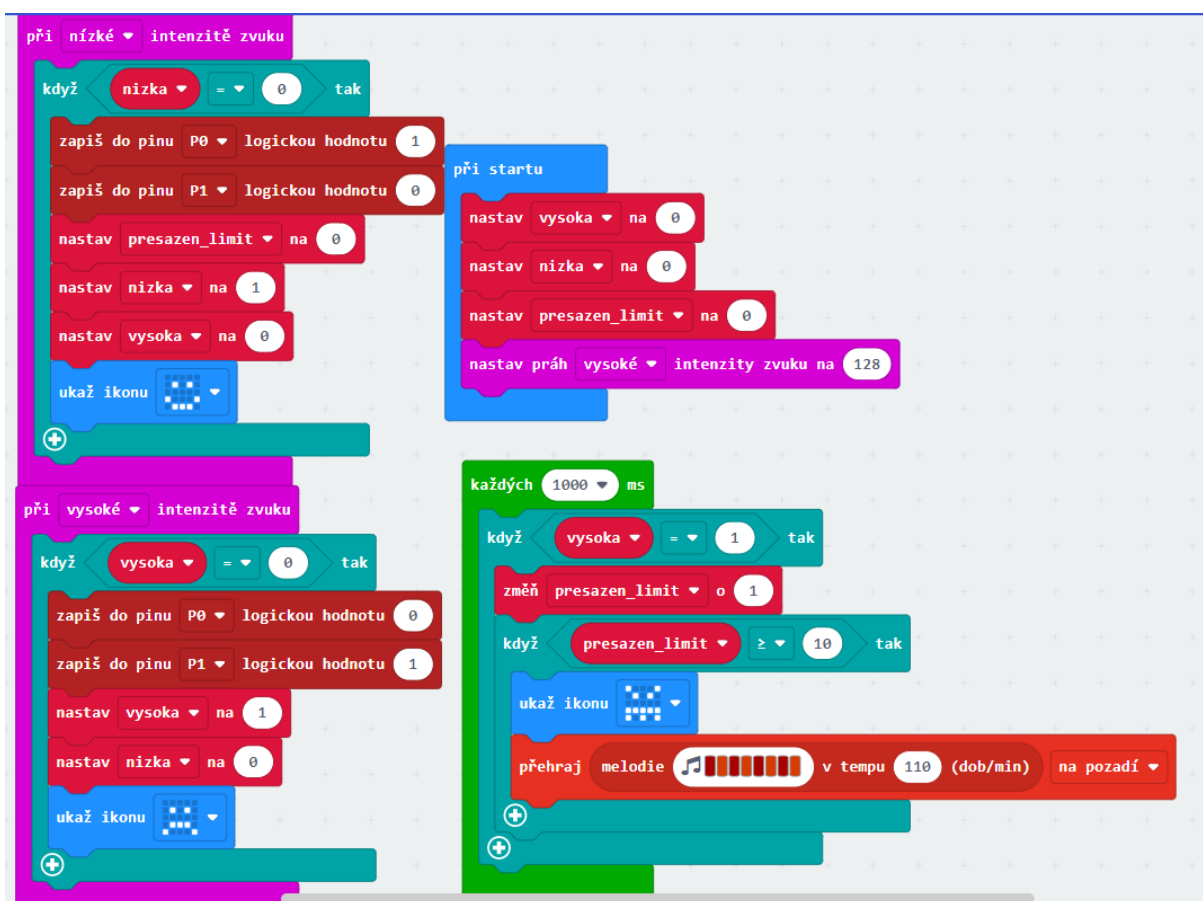
## 6.5. Měřič hluku

Tato úloha využívá mikrofon, který je zakomponovaný v micro:bitu v2. Uvedl jsem jí zde jako ukázkou možného využití funkcí nové verze.

**Zadání:** Vytvořte měřič hluku, který vás varuje v případě přesažení hlasitosti. Využijte micro:bit v2 a diody.

**Poznámka pro učitele:** Tato úloha využívá nových bloků pro měření hluku. Měříme úroveň hluku v okolí a v závislosti na tom svítí dioda. Při přesažení kritické hranice může měřič spustit alarm, který nás varuje, že se máme ztišit, například v případě nočního klidu.

**Cíle a hodnocení:** Tato úloha žáky seznámí s novými funkcemi micro:bitu v2. Žáci využijí nově přidané bloky. Uvědomí si zde rozdíly mezi verzemi a limity micro:bitu v1. Je dobré si ověřit, zda chápou tyto rozdíly a jsou schopni využít nových funkcí.



Obrázek 26: kód pro měřič hluku

**Řešení pro měřič hluku:** V tomto možném řešení využíváme bloky „při nízké/vysoké intenzitě zvuku“. V případě, že je intenzita zvuku nízká, svítí pouze jedna dioda. Pokud přesáhne danou hranici, první dioda zhasne a rozsvítí se druhá. V případě, že je hluk překročen delší dobu, tak micro:bit začne pípat pomocí bzučáku. Intenzitu zvuku si každý může nastavit, jak chce, stejně



tak i varovný systém. Můžeme místo zvuku využít například toho, že budou diody blikat. Je důležité zde upozornit, že bloky „při nízké/vysoké intenzitě zvuku“ se spustí pouze jednou při splnění události a poté až v případě, že intenzita opět změní z vysoké na nízkou či naopak. Proto zde musíme použít opakování, které průběžně kontroluje, jaký je stav. Zároveň máme u každého stavu nastavenou ikonu, která též ukazuje aktuální stav.

**Zapojení:** V tomto případě není nutné nic zapojovat, výhodou micro:bitu je, že má mikrofon i reproduktor již zabudovaný v sobě.

## 7. Odzkoušení v praxi

Po získání potřebných informací, vybrání blokového prostředí a vytvoření těchto příkladových úloh proběhlo samotné odzkoušení v praxi. K tomuto jsem si vybral Střední školu informatiky a právních studií, z. ú. Ač by bylo lepší, kdyby se mi podařilo sehnat odzkoušení v 9. třídě základní školy, zkušenost zde mi stále dala užitečnou zpětnou vazbu. Tuto školu jsem též zvolil hlavně proto, že splňovala to, co jsem potřeboval – žáci měli nějakou zkušenost s micro:bitem nebo alespoň blokovými prostředími, bylo mi poskytnuto více hodin na vyzkoušení a škola vlastní několik micro:bit stavebnic, které jsem mohl použít a které obsahovaly diody, tlačítka, bzučáky a k mému překvapení i breadboard neboli nepájivé pole. Zároveň jsem doufal, že zde budu moci dostat konstruktivní kritiku. Bylo mi poskytnuto 6 vyučovacích hodin, vždy ve dvouhodinových blocích, pro vyzkoušení úloh ve výuce.

Začátek výuky byl chaotický, vzhledem k tomu, že jsem se se třídou neznal. První hodina byla věnována tomu se představit a ujasnit si, jak si představuji průběh výuky. Následně jsem žákům řekl, že téma bude „chytrá domácnost“. Chtěl jsem s nimi společně vymyslet, jaké jsou možnosti a co s micro:bitem můžeme vytvořit. Překvapilo mě, že žákům dělalo poměrně problém pochopit, co se myslí tématem chytrá domácnost, očekával jsem, že jim tento pojem bude známý. Poté, co jsem jim blíže vysvětlil, co se tím myslí, a shrnuli jsme si, jaké bloky máme k dispozici, se žáci rozdělili do skupin a každý si vzal nějaké téma. Nejčastěji žáky napadlo udělat nějaké světlo nebo alarm, navrhl jsem jim možnost vytvořit teploměr a alarm proti vzdálení, aby nedělaly všechny skupiny stejné úlohy. Cílem bylo na konci mít projekty, které by dohromady daly chytrou domácnost. Bohužel se toho víc za první 2 hodiny nestihlo, ale nechal jsem žáky, aby si promysleli, jak budou pracovat na svých projektech příště.

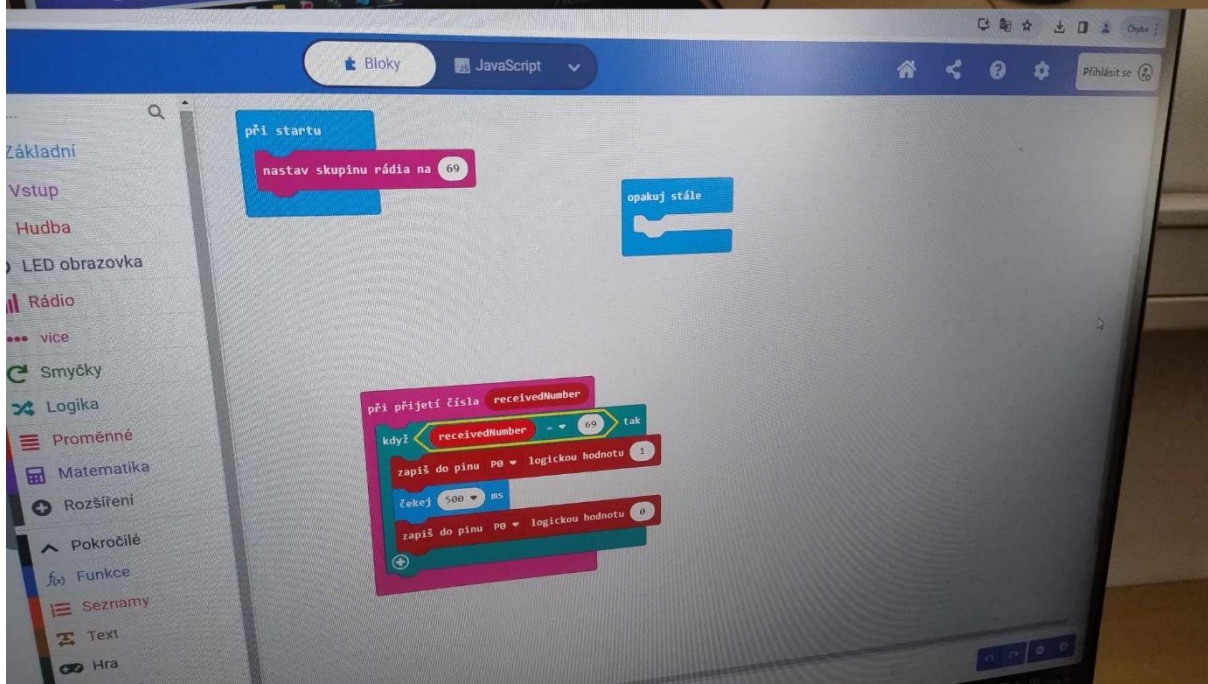
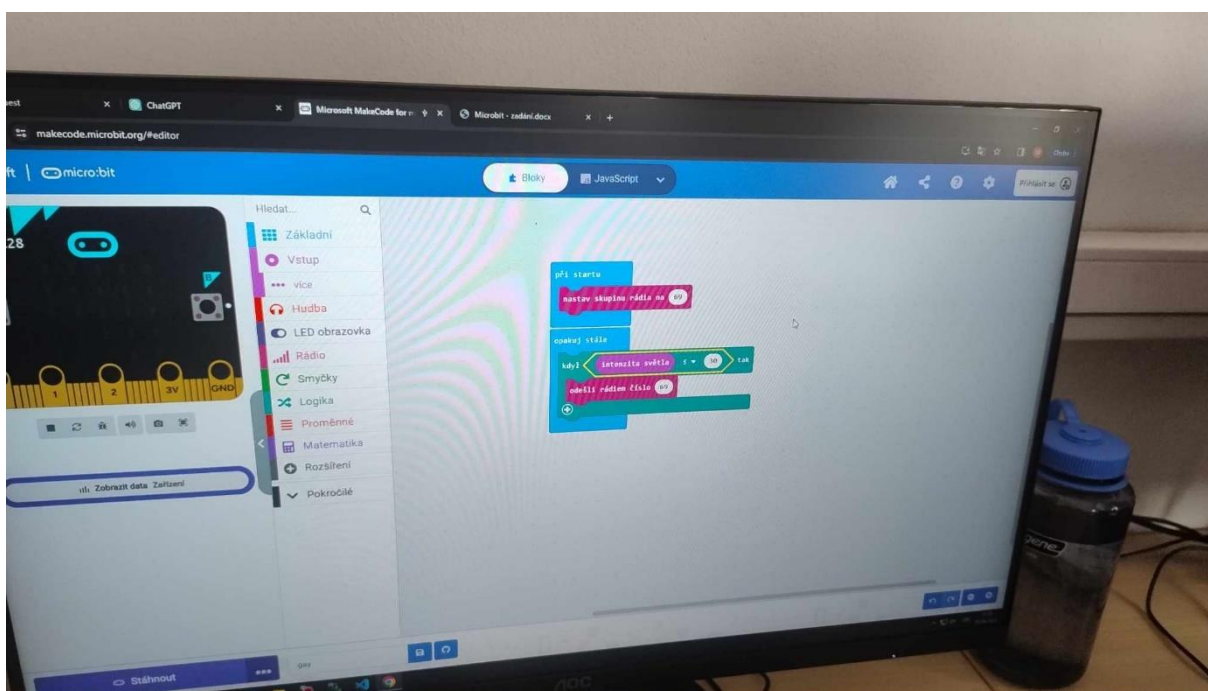


Obrázek 27: žáci začínají pracovat

Následující hodiny jsem žáky nechal pracovat samostatně na tom, co si zvolili. Snažil jsem se do jejich práce příliš nezasahovat, pouze v případě, že si s něčím opravdu nevěděli rady. Osobně se mi líbilo vidět, že některé z možných témat lze pochopit úplně jinak, než jsem předpokládal, a že každý chápe téma nebo přístroj trochu jinak a přistupuje k tomu jinak. Někteří žáci byli rychlejší, tak jsem s nimi potom diskutoval o možných vylepšeních a nedostatcích.



Obrázek 28: testování lampy

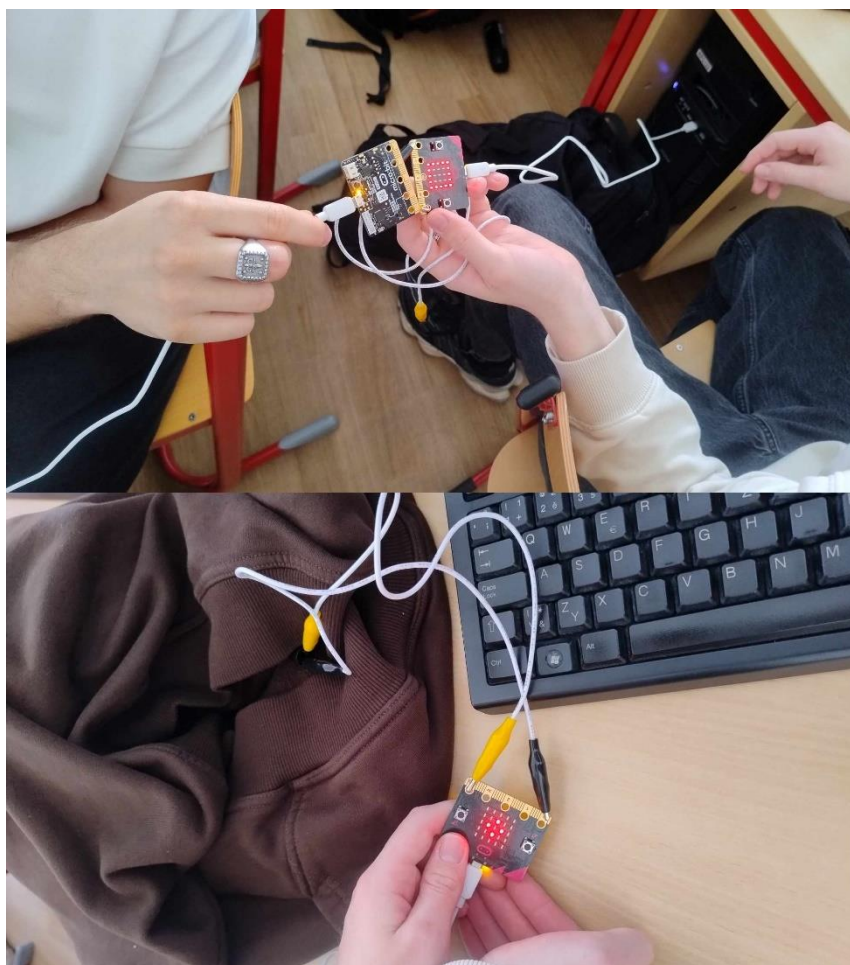


Obrázek 29: upravený kód pro lampu

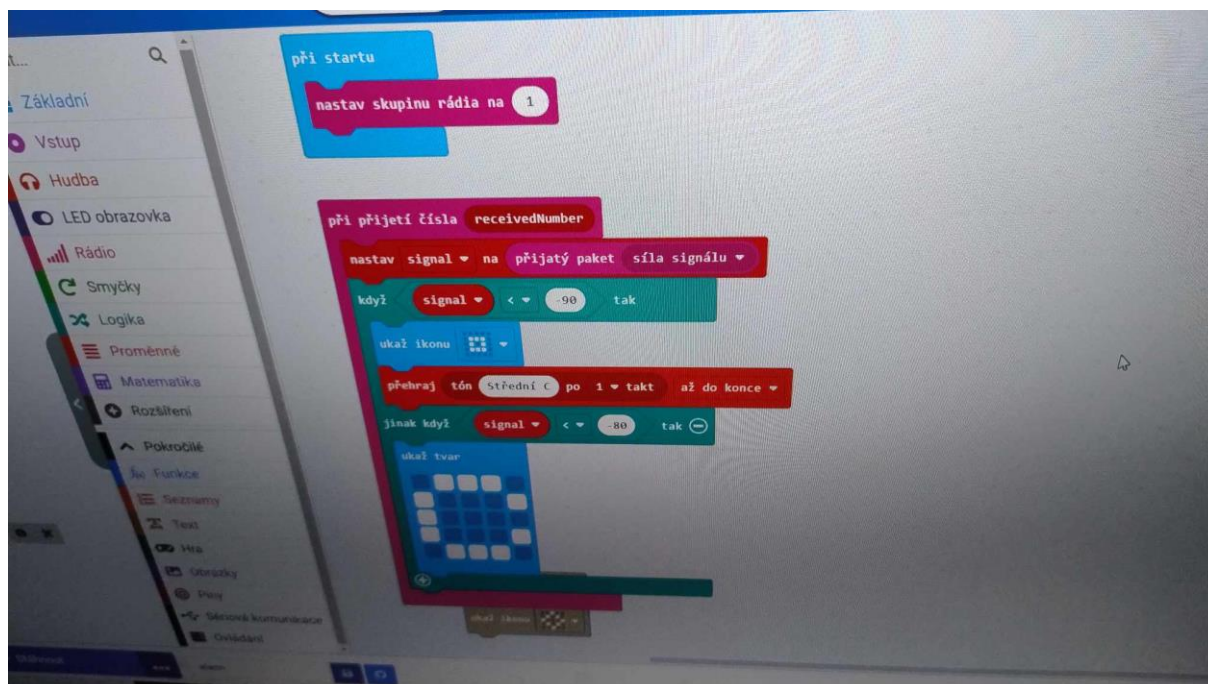


Například jedna skupina žáků, kteří se rozhodli vytvořit lampu, nejdříve přidělali tlačítko na jeden micro:bit a stisknutím rozsvěcovali diodu na druhém. Navrhl jsem jim, zda by nebylo možné udělat rozsvěcování automaticky. To se jim povedlo upravit tak, že zapojili druhý micro:bit a tím posílali zprávy do druhého.

Dále na mě udělala dojem skupina žáků, kteří pracovali na teploměru. Napadlo je nahradit u teploměru bzučák větrákem, tudíž ve chvíli, kdy byla teplota moc vysoká, se větrák spustil, aby začal chladit okolí. Toto mě původně ani nenapadlo vzhledem k tomu, že větráček jsem neměl k dispozici, ale určitě mi to přišlo zajímavější než můj původní způsob s pouhým varovným systémem. Jedné skupině se při vytváření alarmu na dveře povedlo zároveň vytvořit zařízení proti krádeži. Jejich alarm fungoval na základě toho, že měřil sílu signálu mezi 2 micro:bity a při velkém vzdálení začal pípat, což funguje jako alarm na dveře a zároveň i alarm při vzdálení. Toto řešení se mi líbilo, mé původní ukázkové řešení mělo minimální vzdálení na 10 metrů, tudíž se nedalo použít na dveře.



Obrázek 30: žáci zkouší alarm



Obrázek 31: kód alarm / detekce vzdálení od žáků

Bohužel ne všechny skupiny byly úspěšné, některým se nedařilo a i přesto, že jsem se jim snažil pomoci, se jim nepovedlo zařízení zprovoznit. V takovou chvíli přišly vhod předpřipravená řešení, chtěl jsem, aby aspoň ke konci viděli, jak je možné jejich zařízení udělat a mohli si to vyzkoušet. Toto bylo zčásti způsobeno tím, že rychlejší žáci šli do skupiny společně a pomalejší žáci byli ve skupině též společně, tudíž byl mezi nějakými skupinami poměrný rozdíl. Vzhledem k tomu, že jsem třídu osobně neznal, tak jsem toto zpozoroval až příliš pozdě, ale věřím, že v případě, kdy je učitel ve třídě delší dobu, by si toto mohl jednoduše ohlídat a postarat se, aby skupiny byly více vyvážené.

Z důvodu nedostatku času se již nestihlo vytvořená a zapojená zařízení vložit do vytvořené konstrukce a finalizovat tak výrobky. Věřím ale, že se stihlo otestovat to nejdůležitější, a to je programování a zapojení samotných micro:bitů s periferiemi.

### 7.1. Zpětná vazba

Posledních pár minut jsem využil k získání zpětné vazby od žáků. Zajímalo mě, jestli je bavila práce s micro:bitem a vybrané téma. Většina třídy říkala, že je to bavilo, někteří preferovali tento styl výuky a toto téma více než klasickou výuku. Dále jsem chtěl vědět, jestli jim to, co vytvářeli, přišlo užitečné, odpověď byla kladná, dokázali by si představit nějakou takovou vychytávku mít doma. Největším problémem byla pro žáky technická limitace micro:bitu a periférií, které měli k dispozici. Určitě by bylo dobré mít na takovouto výuku pokročilejší stavebnice, aby žáci nemuseli využívat pouze základních senzorů. Též jim některé možnosti

v blokovém programování přišly limitující. Na otázku, co by změnili, mi žáci překvapivě odpověděli, že by se jim líbilo mít nějaké podrobnější a přímočařejší zadání úloh. Někteří říkali, že by se jim více líbilo mít jasnější zadání a u složitějších kódů mít například k dispozici nápovědu. Taková úprava by se však vzdalovala od konceptu projektové výuky, i když věřím, že některým žákům by to vyhovovalo více. Na samotný Microsoft Makecode nebylo mnoho specifických stížností, jediná častější stížnost byla na to, že některé popisy bloků mají kostrbatější překlad do češtiny. Jako poslední bylo to, že žáci chtěli jasnější popis k tomu, co je to vlastně chytrá domácnost. Ze zkušenosti z první hodiny jsem sám zjistil, že ne každému je jasné, co tento koncept znamená, a bylo by vhodné ho detailněji vysvětlit jako přípravu na tuto projektovou výuku.

## 7.2. Úpravy na základě zpětné vazby

### 7.2.1. Úvod

Na základě zpětné vazby, kterou jsem dostal, a zároveň toho co jsem vypořezoval během testování, jsem se rozhodl provést úpravy na daných úlohách. První důležitou úpravou je samotný úvod do zadání. Zjistil jsem, že je nutné žáky více obeznámit s konceptem chytré domácnosti. Proto jsem se rozhodl přidat následující odstavec jako úvod:

*V dnešní době existuje spousta chytrých zařízení, které nám ulehčují život v domácnosti. Můžeme si pořídit lampu, nebo dokonce i konvici, kterou ovládáme telefonem. Máme alarmy, které jsou schopné zaznamenat, že někdo přišel, anebo například termostaty, které udržují teplotu v místnosti. Dnes se dá pořídit i chytrá lednice, která vám sděluje její obsah a doporučuje recepty. Veškerá tyto zařízení mají jedno společné, a to je, že používají mikropočítač pro jejich funkce. Takový mikropočítač před sebou máte nyní i vy a pomocí něj si můžete vytvořit vlastní zařízení do chytré domácnosti. Jaká taková zařízení vás napadají?*

Zároveň je dobré každé skupině žáků určit skupinu rádia, kterou mají využívat, aby pak nedocházelo k tomu, že se skupiny budou navzájem překrývat, což může způsobit nečekané problémy.

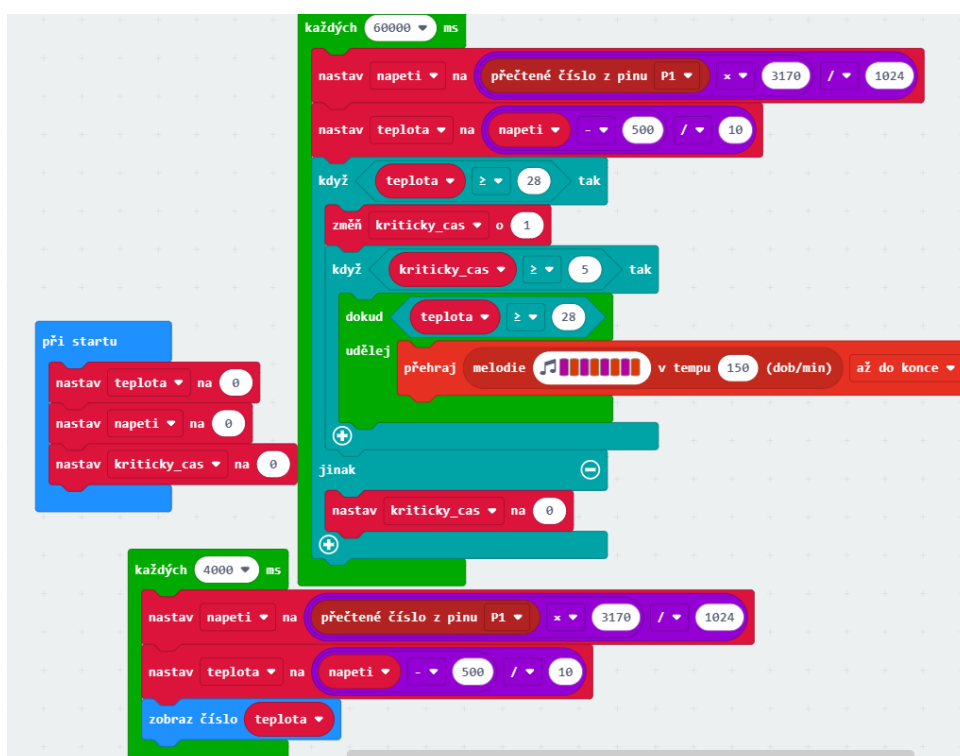
### 7.2.2. Automatická lampa

Žáci, kteří vytvářeli lampu měli často problém s tím, že jejich lampa nesvítla pořád. To bylo způsobeno tím, že žáci nechápali, proč využívat druhý micro:bit, a světlo i senzor měli v jednom. To způsobilo, že micro:bit detekoval světlo a vypnul se, jen aby se poté znovu zapnul. V případě, že se toto žákům stane ve výuce, je dobré jim například navrhnout, že by se světlo mělo zapínat na základě světla venku.

Zadání bych tedy upravil takto: „Vytvořte automatickou lampu, která se bude zapínat a vypínat na základě venkovního detektoru světla. Použijte alespoň 2 micro:bity a diodu.“

### 7.2.3. Teploměr

Úloha teploměru byla úspěšná, líbil se mi nápad použít větráček místo bzučáku, je to zajímavější a alespoň to řeší problém, který to naměří místo pouhého varování. Žáci chtěli využít externí teploměr, což se jim bohužel nepovedlo zprovoznit, a ani já jsem nebyl připraven na to, jak z něj správně měřit. Ovšem je lepší využít externí teploměr místo interního vzhledem k tomu, že interní měří teplotu CPU, která nemusí být přesná. Zároveň to dává žákům komplexnější problém a zkušenost s tím, jak takový senzor může fungovat. Rozhodl jsem se tedy ukázkové řešení pozměnit a využít v něm externí teploměr místo interního. Vzoreček k výpočtu může být u různých teploměrů jiný, je tedy důležité zkontrolovat pomocí datasheetu daného teploměru. Většina kódu tedy vypadá podobně, jen je zde změna ve formě přidání dvou bloků pro výpočet teploty na začátek. Vzoreček pro mnou použitý teploměr jsem našel v učebnici „ROBOTIKA PRO ZÁKLADNÍ ŠKOLY: PROGRAMUJEME MICRO:BIT POMOCÍ MAKECODE“ [1]. Samotný vzoreček pro výpočet doporučuji žákům ukázat vzhledem k tomu, že může být pro jejich chápání moc složitý a zároveň jeho pochopení není důležité pro splnění úlohy.

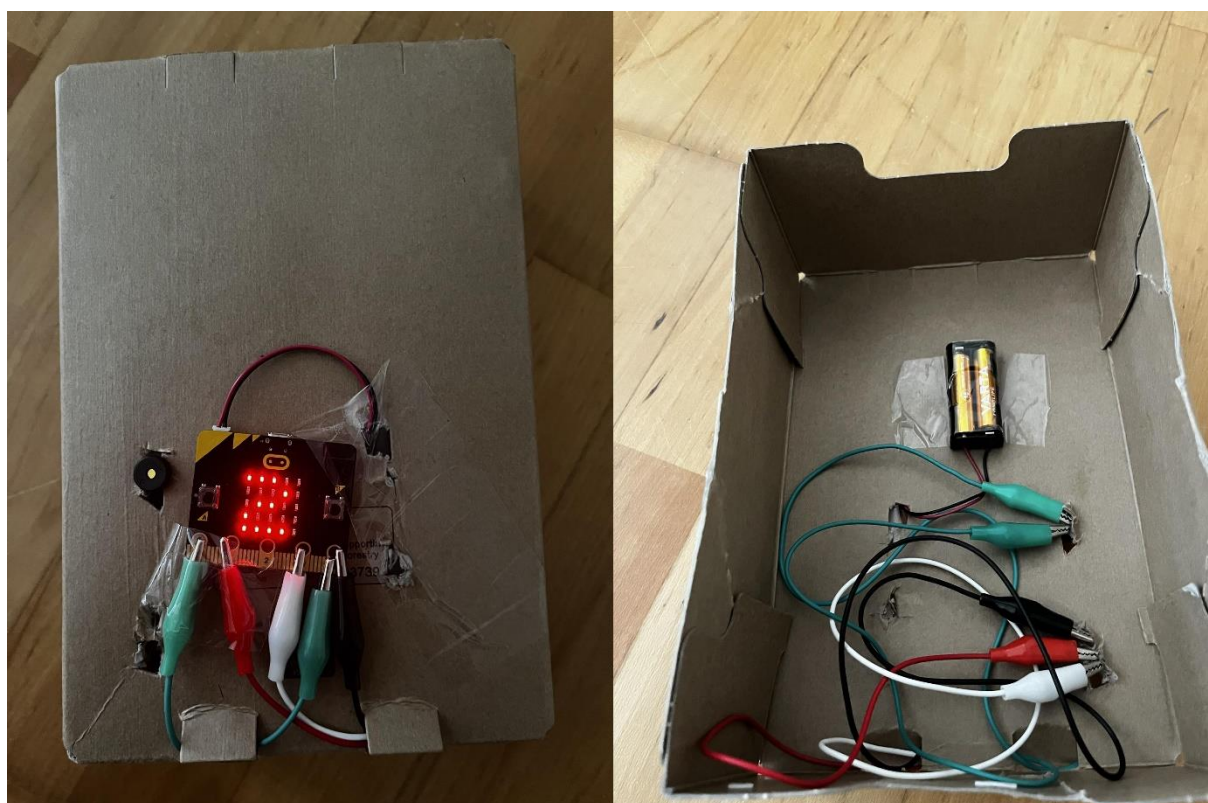


Obrázek 32: upravený kód pro teploměr



Zapojení se změní takto:

**Zapojení a produkt:** Důležité je zde především správné zapojení samotného teploměru. Pokud je položen plochou částí nahoru, tak pin vlevo je „Vin“, který zapojíme na micro:bitu do 3 V. Prostřední pin je „Vout“, tento pin zapojíme do námi zvoleného pinu, například P1. Dále stačí zapojit bzučák, kde opět jeden pin půjde do námi zvoleného pinu, například P2, a druhý zapojíme do GND. K tvorbě výsledného produktu jsem využil krabici, do které jsem udělal díry tak, abych ní mohl provléct přípojky. Vzhledem k tomu, že chceme vidět teplotu, micro:bit musí být přilepen navrchu. Výsledný produkt tedy může vypadat například takto:

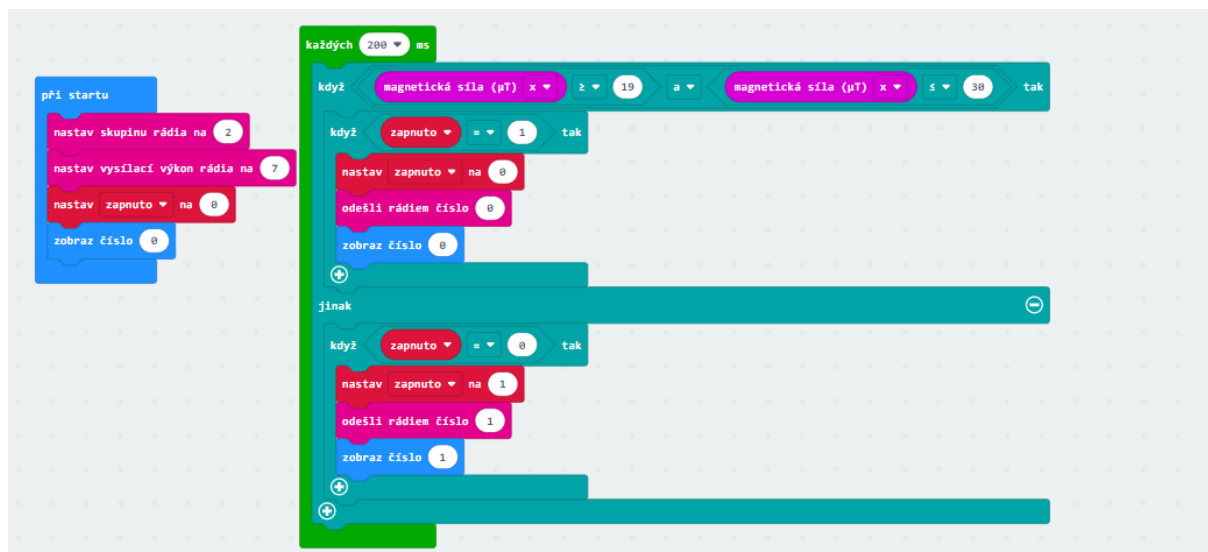


Obrázek 33: upravené zapojení teploměru

#### 7.2.4. Alarm na dveře

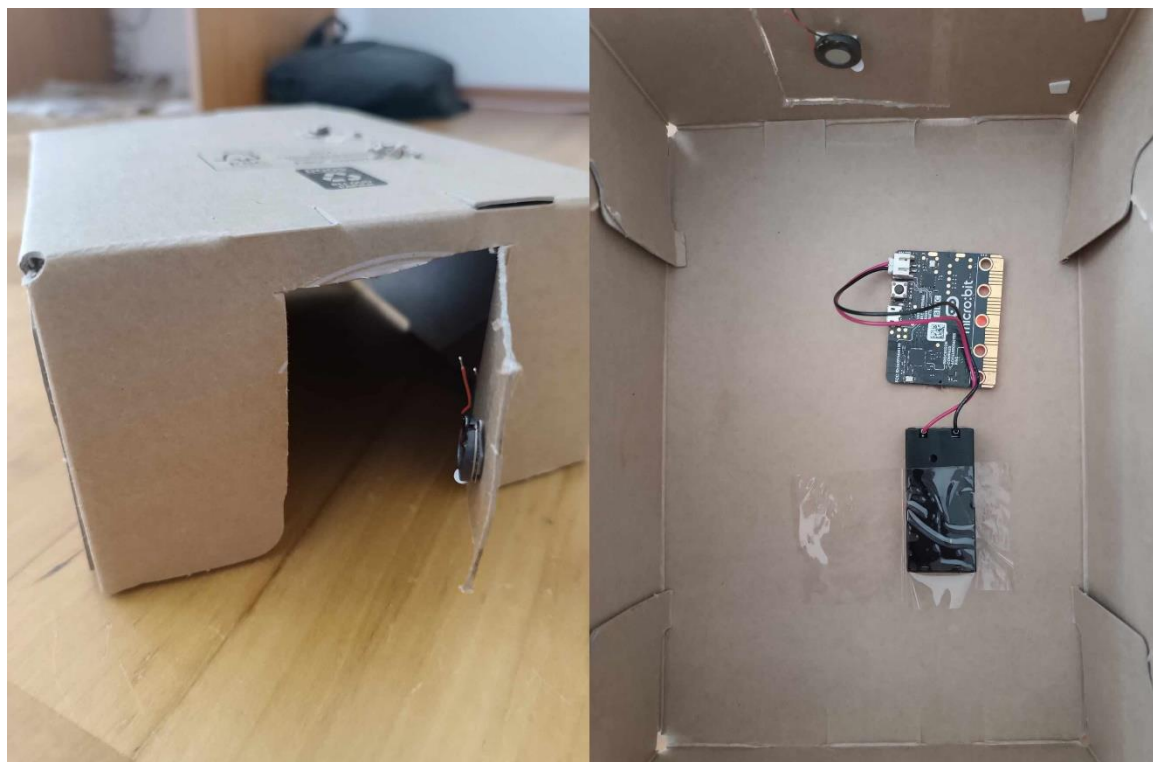
Alarm na dveře byl poměrně problémová úloha. Jediné řešení bylo měření vzdálenosti mezi 2 micro:bity, které popisují v další kapitole. Osobně mi toto řešení nepřijde ideální, jelikož vyžaduje mít na dveřích 2 micro:bity, kdy pak samotný alarm musí tedy být buď též na dveřích, anebo se musí využít 3. micro:bit. Snažil jsem se žákům poradit, aby využili kompas, ale to nebylo moc užitečné a nedařilo se jim to zprovoznit. Zároveň byl kompas nekonzistentní a kalibrace byla zdlouhavá. Rozhodl jsem se tedy v ukázkovém řešení místo kompasu použít

magnet. Micro:bit je schopný měřit okolní magnetické pole a toho využívá i toto řešení. Na rám dveří připevníme magnet a na dveře micro:bit či naopak.



Obrázek 34: upravený kód pro alarm

Kód je podobný předchozí verzi, jen zde místo měření kompasu používám měření magnetické síly. Kód pro druhý micro:bit zůstává stejný. Výsledný produkt ovšem bude vypadat jinak, vzhledem k tomu, že používáme magnet, tak je nutné toto zakomponovat do výsledného produktu. Opět jsem si vytvořil provizorní dveře z krabice. Na dveře jsem přilepil magnet a vedle rámu micro:bit. Výsledek tedy může vypadat například takto:



Obrázek 35: upravené dveře

### 7.2.5. Zabezpečení proti krádeži

Při vytváření alarmu na dveře se žákům povedlo udělat kód, který funguje tak, jak by měla fungovat tato úloha, ale ještě lépe, než jsem zamýšlel. V rámci alarmu na dveře se snažili přijít na to, jak zjistit vzdálenost dvou micro:bitů a použili blok na měření síly signálu. Díky tomuto je možné měřit vzdálenost mezi micro:bity přesněji a rozhodl jsem se tak upravit ukázkové řešení.



Obrázek 36: upravené řešení pro zabezpečení proti krádeži

Zde je vidět, že používám proměnnou „přijatý paket síla signálu“. Tato proměnná mi měří, jak silný signál přišel od druhého micro:bitu, díky čemuž jsem schopen vnímat jejich vzdálenost. Proto spustím alarm v tu chvíli, co vzdálenost přesáhne nějakou hranici. Pomocí vysílacího výkonu si můžeme nastavit přesnost tohoto měření. Na kratší vzdálenosti je lepší použít menší výkon, kdy je pak micro:bit schopen zaznamenat pohnutí každých pár centimetrů. Tento kód je díky tomuto přesnější, přehlednější a zároveň si díky němu můžeme přesněji určit vzdálenost, do jaké se micro:bity nesmí vzdálit.

### 7.2.6. Shrnutí

Ve výsledku jsem spokojen se zpětnou vazbou, kterou mi testování dalo. Věřím, že ukázkové úlohy jsou nyní lepší a zároveň jsou bližší tomu, co by se dalo v praxi čekat. Dostal jsem jiné pohledy na věc a setkal se s problémy, které jsem neočekával. Nejvíce lituji toho, že jsem podcenil časovou dotaci. Pro použití v praxi je určitě nutné více času, aby se vše dalo v klidu stihnout. Jsem rád, že žáky úlohy bavily a přišly jim zajímavé, mě samotného práce s micro:bitem též bavila a věřím, že je vhodný do výuky díky své kombinaci jednoduchosti a dostupných možností. Určitě bych ho rád využil ve své budoucí výuce.

## 8. Závěr

Výsledkem celé práce jsou úlohy určené k projektové výuce s micro:bitem. Při vytváření úloh bylo zvoleno a popsáno několik blokových programovacích prostředí pro práci s destičkou micro:bit. Následně bylo provedeno porovnání možných blokových prostředí, kde je zaměřeno na to, jak vhodné je blokové prostředí pro destičku micro:bit. Bylo vybráno blokové prostředí Microsoft Makecode vzhledem k tomu, že bylo zhodnoceno jako nejvíce vyhovující jak pro destičku micro:bit, tak pro žáky. Samotná destička je zde též popsána, zejména jsou důležité rozdíly mezi verzí v1 a v2. Zároveň jsou zde popsány základní zásady projektové výuky. Dále jsou zde popsány internetové stránky, kde je možné se inspirovat pro další úlohy využívající destičku micro:bit a programovací prostředí Microsoft Makecode.

Byly vytvořeny úlohy využívající programovací desku micro:bit určené pro druhý stupeň základní školy. Každá úloha využívá periferie k micro:bitu, žáci tak krom pouhého programování micro:bitu i zapojují a programují samotné periferie. Úlohy počítají s předchozími znalostmi a jsou určeny pro pokročilejší žáky. Úlohy jsou koncipovány jako možná řešení při projektové výuce na téma „Chytrá domácnost“. Tyto úlohy byly následně vyzkoušeny ve výuce s pokročilejšími žáky. Bylo mi líto, že se úlohy nestihly řádně otestovat v 9. třídě s dostatečným množstvím času, ale i přesto jsem byl se zpětnou vazbou spokojen. Díky zpětné vazbě jsem zjistil, jaké jsou nedostatky vytvořených úloh, přesněji jsem určil co očekávat a zároveň si uvědomil, že jim může být koncept chytré domácnosti cizí. Zároveň jsem díky sledování žáků při práci získal jiný úhel pohledu. Na základě zpětné vazby a pozorování byla tedy každá úloha upravena, aby více odpovídala tomu, co se dá při výuce očekávat, a zároveň aby neměla zbytečně komplikované řešení. Ke každé úloze je přiložena poznámka pro učitele, kde je detailněji popsáno, co se v úloze očekává, možné zadání pro žáky a poté cíle, kterých úloha dosahuje. Zároveň je k nim přidáno možné řešení v Microsoft Makecode a poté možné zapojení do výsledného produktu.

Micro:bit, společně s blokovým prostředím Microsoft Makecode, mi přišly vhodné do výuky, obojí mi vyhovovalo a nasetkal jsem se s velkou limitací. I reakce žáků byla pozitivní, téma jim přišlo zajímavé a jejich výrobky užitečné.

## Seznam použité literatury a zdrojů

- [1] PECH, Jiří; PRŠALA, Jan; VANÍČEK, Jiří a NOVÁK, Milan. *Robotika pro základní školy: programujeme micro:bit pomocí Makecode*. 2021.
- [2] BH, . *Microbiti*. Dostupné také z: <https://www.microbiti.cz/>.
- [3] HALFACREE, Gareth. *The official BBC micro:bit user guide*. John Wiley et Sons, Indianapolis, IN, 2018, 2018.
- [4] *Microbit*. Microbit. Dostupné také z: <https://microbit.org>.
- [5] AUSTIN, Jonny; BAKER, Howard; BALL, Thomas; DEVINE, James; FINNEY, James et al. *Communications ACM*. In: The BBC micro:bit – From the U.K. to the World. 2020. Dostupné také z: <https://cacm.acm.org/research/the-bbc-microbit-from-the-u-k-to-the-world/#R7>.
- [6] IW, . *Bbb.co.uk*. Microbit-first-year. 2017. Dostupné také z: <https://www.bbc.co.uk/mediacentre/latestnews/2017/microbit-first-year>.
- [7] BRICKHACKER, EMMET. *Bastlírna HW Kitchen*. BBC MICRO:BIT V2 UŽ KONCEM ROKU 2020. 2020. Dostupné také z: [https://bastlirna.hwkitchen.cz/bbc-microbit-v2-uz-koncem-roku-2020/#Co\\_prinasi\\_micro\\_bit\\_v2\\_noveho](https://bastlirna.hwkitchen.cz/bbc-microbit-v2-uz-koncem-roku-2020/#Co_prinasi_micro_bit_v2_noveho).
- [8] MICRO:BIT EDUCATIONAL FOUNDATION. *Micro:bit Help & Support*. BBC micro:bit V2 interface processor change (V2.2). 2022. Dostupné také z: <https://support.microbit.org/support/solutions/articles/19000132336-bbc-micro-bit-v2-interface-processor-change-v2-2->.
- [9] MAJED, Marji. *Learn to Program with Scratch: A Visual Introduction to Programming with Games, Art, Science, and Math*. No Starch Pr, 2014, s. 288.
- [10] VITTASCIENCE, . *Vittascience*. Vittascience. 2024. Dostupné také z: <https://en.vittascience.com/>.
- [11] ZORMANOVÁ, Lucie. *Projektová výuka*. Metodický portál RVP. 2012. Dostupné také z: <https://clanky.rvp.cz/clanek/c/Z/14983/projektova-vyuka.html>.
- [12] LOUŽECKÁ, Iva. *Projektová výuka nejen v současné situaci*. Ve škole. 2020. Dostupné také z: <https://www.veskole.cz/clanky/projektova-vyuka-nejen-v-soucasne-situaci>.
- [13] MAZÁČOVÁ, Nataša. *Možnosti a meze projektové výuky v současné škole*. Metodický portál RVP. 2007. Dostupné také z: <https://clanky.rvp.cz/clanek/c/z/1288/MOZNOSTI-A-MEZE-PROJEKTOVE-VYUKY-V-SOUCASNE-SKOLE.html>.
- [14] KUBÍNOVÁ, Marie. *Projekty ve vyučování*. In: Metodický portál RVP. 2005. Dostupné také z: <https://clanky.rvp.cz/clanek/c/z/334/PROJEKTY-VE-VYUCOVANI.html>.
- [15] KADLEČÍKOVÁ, Jana. *LEGO Mindstorms Ev3 v projektové výuce na střední škole*. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2015. Dostupné také z: <https://theses.cz/id/3uvmkd/>.
- [16] MAŇÁK, Josef a ŠVEC, Vlastimil. *Výukové metody*. 1. Paido, 2003, s. 219.
- [17] MICROSOFT. *Microsoft Makecode*. Microsoft Makecode. Dostupné také z: <https://makecode.microbit.org/>.

- [18] UCIMESROBOTY. *Archiv rubriky: BBC micro:bit*. Učíme s roboty. Dostupné také z: <https://ucimesroboty.cz/index.php/category/bbc-microbit/>.
- [19] TEACH WITH IT. *MICRO:BIT PHYSICAL COMPUTING GUIDES*. Teach with IT. Dostupné také z: <https://www.teachwithict.com/pushbtton.html>.

## Použité obrázky

Obrázek 1: schéma Micro:bit .....	13
Obrázek 2: rozdíly mezi verzí 1.5 a 2 [7].....	14
Obrázek 3: rozhraní Scratche .....	15
Obrázek 4: rozhraní Python editor .....	16
Obrázek 5: rozhraní Microsoft Makecode .....	17
Obrázek 6: možné tutoriály u Makecode Microbit.....	18
Obrázek 7: rozhraní Vittascience.....	19
Obrázek 8: kategorie projektů u Microsoft Makecode [17] .....	21
Obrázek 9: test šikovnosti na stránce Microbiti [2].....	22
Obrázek 10: příklad ze stránky Učíme s roboty [18].....	23
Obrázek 11: projekty na stránce Microbit [4] .....	24
Obrázek 12: popis projektu na Microbit [4].....	25
Obrázek 13: kód pro senzor světla .....	30
Obrázek 14: kód pro lampu .....	31
Obrázek 15: zapojení diody .....	31
Obrázek 16: produkt lampa.....	32
Obrázek 17: lampa se stínidlem .....	33
Obrázek 18: produkt senzor světla .....	33
Obrázek 19: kód pro teploměr.....	34
Obrázek 20: kód pro senzor otevření.....	35
Obrázek 21: kód pro alarm.....	36
Obrázek 22: zapojení tlačítka [19] .....	37
Obrázek 23: dveře pro alarm.....	38
Obrázek 24: produkt alarm .....	38
Obrázek 25: kód pro zabezpečení proti krádeži .....	39
Obrázek 26: kód pro měřič hluku .....	40
Obrázek 27: žáci začínají pracovat.....	43
Obrázek 28: testování lampy.....	44
Obrázek 29: upravený kód pro lampu .....	44
Obrázek 30: žáci zkouší alarm .....	45
Obrázek 31: kód alarm / detekce vzdálení od žáků.....	46
Obrázek 32: upravený kód pro teploměr.....	48
Obrázek 33: upravené zapojení teploměru .....	49
Obrázek 34: upravený kód pro alarm.....	50
Obrázek 35: upravené dveře .....	50
Obrázek 36: upravené řešení pro zabezpečení proti krádeži.....	51



## **Přílohy**

### **Příloha 1 – Zadání úloh**

Uvedená příloha je též dostupná na na adrese: <https://tinyurl.com/yc7jjxtk>

## Úvod

*V dnešní době existuje spousta chytrých zařízení, které nám ulehčují život v domácnosti. Můžeme si pořídit lampu, nebo dokonce i konvici, kterou ovládáme telefonem. Máme alarmy, které jsou schopné zaznamenat, že někdo přišel, anebo například termostaty, které udržují teplotu v místnosti. Dnes se dá pořídit i chytrá lednice, která vám sděluje její obsah a doporučuje recepty. Veškerá tyto zařízení mají jedno společné, a to je, že používají mikropočítač pro jejich funkce. Takový mikropočítač před sebou máte nyní i vy a pomocí něj si můžete vytvořit vlastní zařízení do chytré domácnosti. Jaká taková zařízení vás napadají?*

### Poznámka pro učitele

Cílem je vyzkoušet, zda jsou žáci schopni naprogramovat i zapojit své vlastní zařízení v rámci tématu „Chytrá domácnost“. Veškeré úlohy zde využívají blokového programovacího prostředí Microsoft Makecode. Počítá se s tím, že žáci mají pokročilejší znalosti práce s destičkou. Pro finální produkt doporučuji, aby si žáci přinesli krabičku, do které mohou následně micro:bit vložit a vytvořit díky ní tak kompletní produkt. Na začátku je důležité se žáky provést „brainstorming“ a vymyslet, jak mohou daná zařízení a produkty vypadat. Pro inspiraci je zde několik ukázkových úloh, které lze v rámci projektové výuky stvořit.

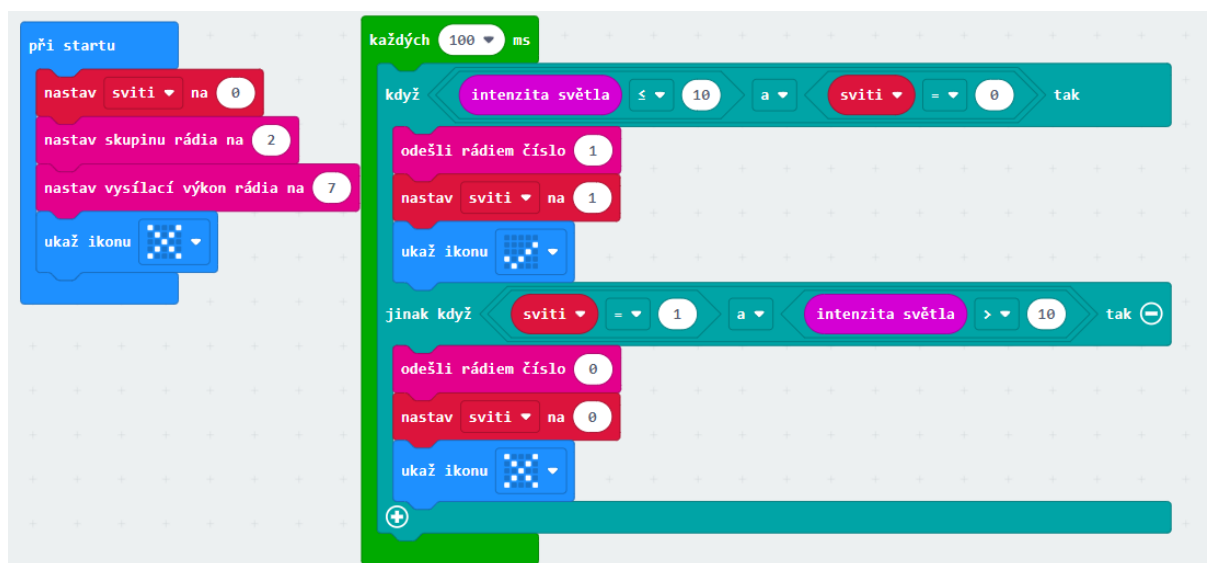
## Automatická lampa

**Zadání:** Vytvořte automatickou lampu, která se bude zapínat a vypínat na základě venkovního detektoru světla. Použijte 2 micro:bity a diodu.

**Poznámka pro učitele:** Úlohou je automatická lampa do domácnosti – lampa, která detekuje světlo venku a v moment, kdy je tma, se rozsvítí. K sestavení této úlohy jsou potřeba 2 micro:bity a dioda. Jeden měří světlo a vysílá signál k zapnutí či vypnutí. Druhý je napojen na lampu, v našem případě na LED diodu, a po přijetí signálu ji rozsvítí. Poskytuji zde možné řešení, vše je však na žácích a mohou přijít s úplně jiným způsobem řešení.

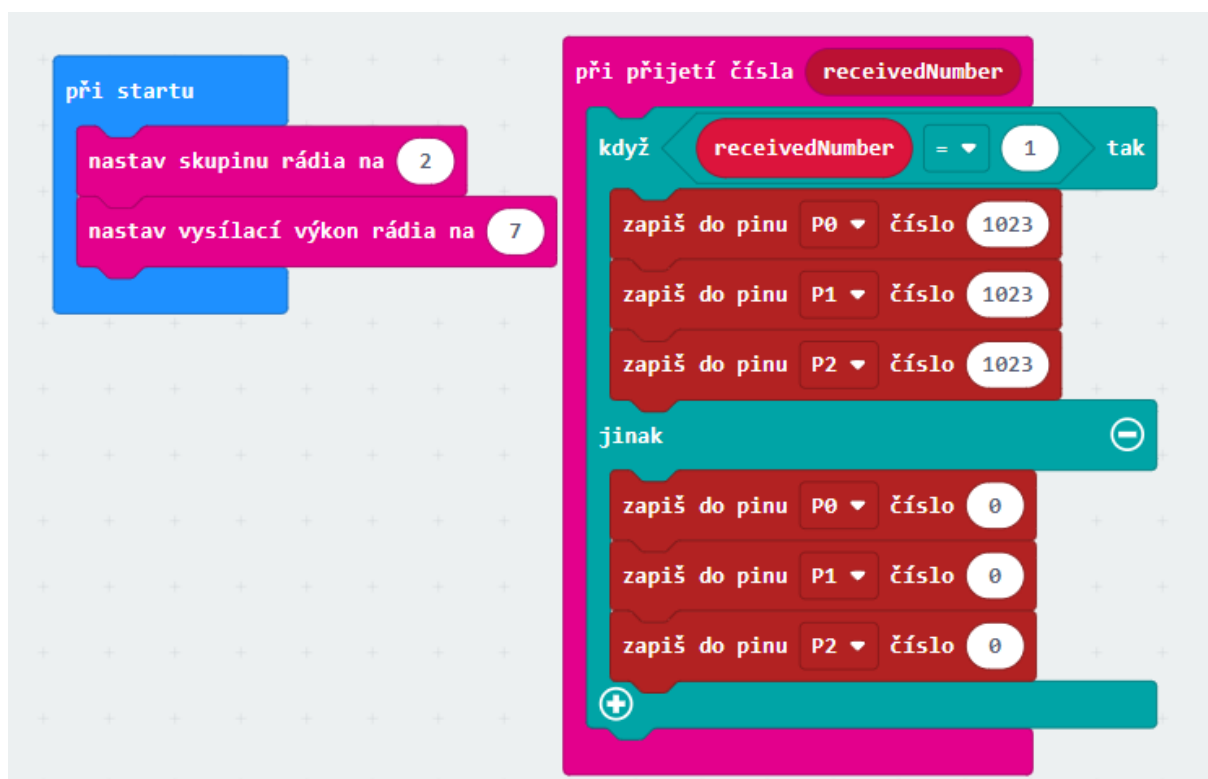
**Cíl a hodnocení:** Žáci si ověří svou schopnost zapojit diodu a pracovat s piny. Dále žáci využijí, bloky rádia a senzor světla zabudovaný v micro:bitu. Je důležité pozorovat, zda žákům dojde, že při použití jednoho micro:bitu se lampa bude zapínat a vypínat dokola, protože bude vnímat svoje světlo, a proto potřebujeme alespoň 2 ke správnému fungování.

**Řešení pro senzor světla:** Jako první je nutné si sestavit samotný senzor světla. Využijeme vestavěného senzoru světla v micro:bitu. Dále je důležité chápat fungování bloků rádia.

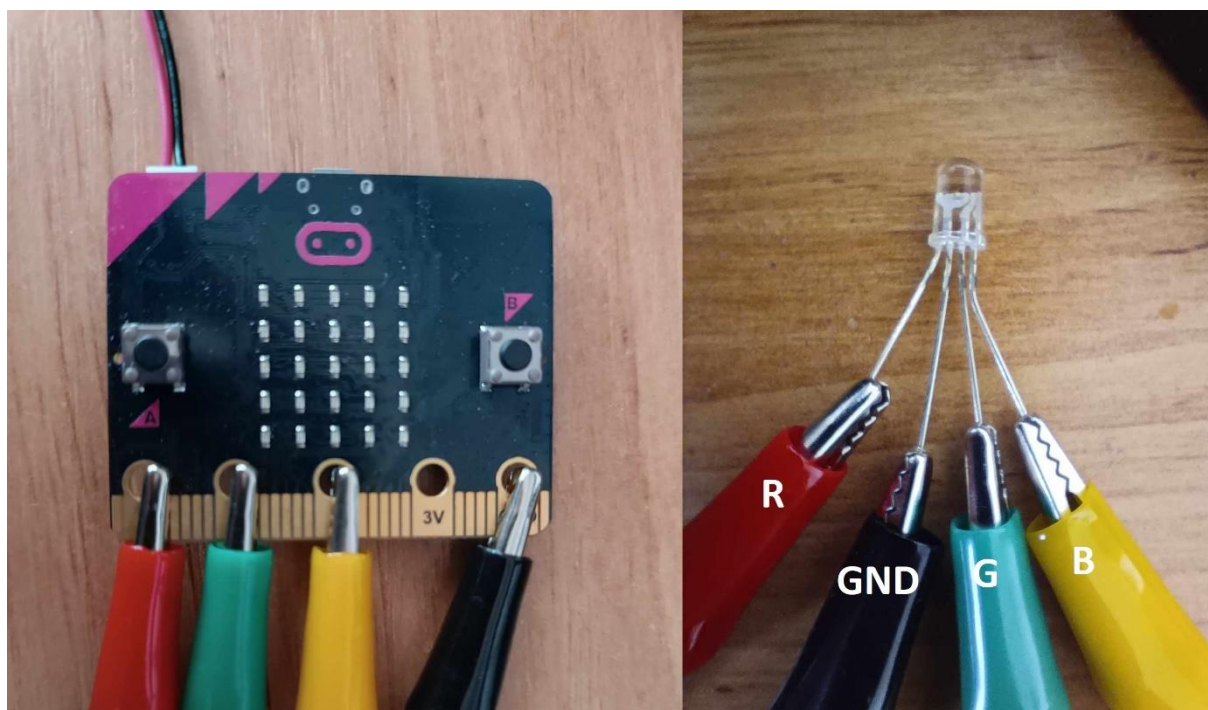


Zde je mnou navrhnutý kód pro senzor. Při startu se nastaví skupina rádia, zobrazí se ikona, že lampa je vypnutá, a nastaví se proměnná svítí. Tato proměnná nám určuje, zda je lampa zapnutá či ne. Následně se každých 100ms ověří, zda je v okolí dostatečné světlo, a v případě, že okolní světlo padne pod určitou hranici, vyšle se signál 1 a změní se ikona. Zároveň se ověřuje, zda již lampa není zapnutá, aby se neposílal signál zbytečně dokola.

**Řešení pro lampu:** U lampy připojíme LED diodu, můžeme vybrat normální či RGB. Pin si můžeme vybrat, jaký chceme, záleží na typu diody. Důležité je správně určit katodu a tu zapojit do GND.

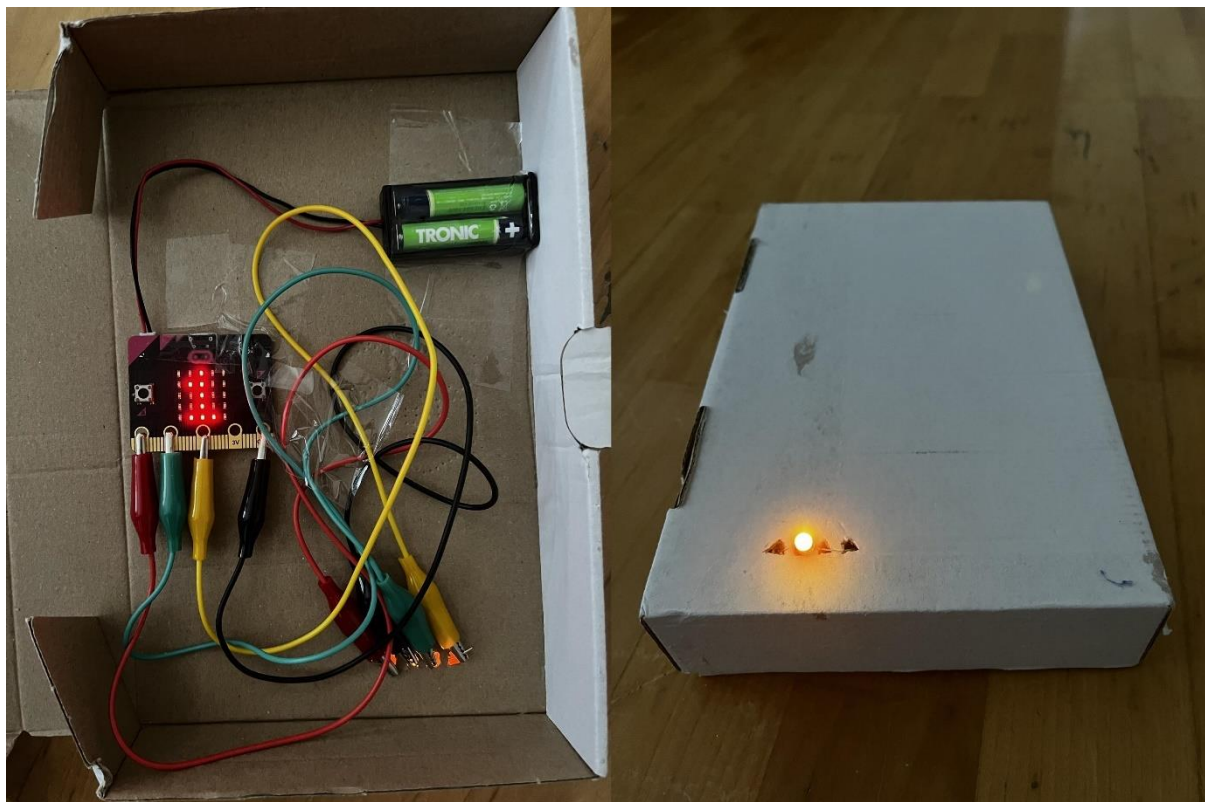


Toto je možný kód pro samotné zapínání. Používáme zde RGB diodu. Na začátku si tedy nastavíme správnou skupinu rádia a poté po přijetí čísla kontrolujeme, jaké číslo jsme dostali.



V případě, že přišlo číslo 1, tak se všechny piny zapnou a dostaneme tím bílé světlo. V případě přijetí jakéhokoliv jiného čísla se všechny piny nastaví na hodnotu 0.

**Zapojení a produkt:** U zapojení je důležité dát si především pozor na správné zapojení RGB diody. Nejdelší zakřivená část je anoda, ta přijde na micro:bitu do části GND. Zbytek jsou barvy, červená, zelená a modrá. Následně můžeme využít například kartónové krabice, do které celý micro:bit schováme a uděláme díry pro diodu. Výsledný produkt tedy může vypadat například takto:



Pokud chceme, můžeme si k diodě přidělat i vlastní stínidlo pomocí papíru. Stačí si z papíru udělat válec, nalepit navrch kolečko z papíru a přilepit to k naší diodě. Může to ve výsledku vypadat například takto:



Samotný senzor musí být venku, není tedy dobré ho schovávat, může vypadat například takto:



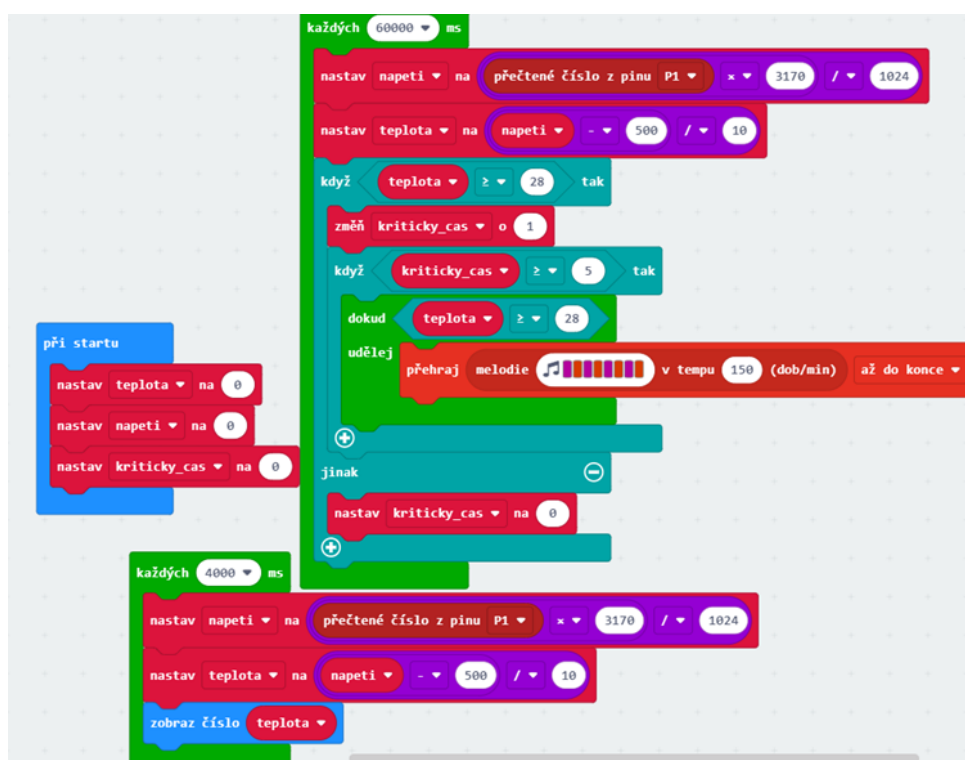


## Teploměr

**Zadání:** Vytvořte teploměr, který ukazuje teplotu a v případě přesážení limitu spustí varování. Využijte alespoň 2 periferie.

**Poznámka pro učitele:** Další možný projekt pro žáky je teploměr. Využijeme senzoru teploty, který si připojíme. Žáci mohou buď využít dvou micro:bitů a měřit díky tomu teplotu venku. Toto má tu výhodu, že můžeme využít micro:bit z předchozí úlohy a nechat ho posílat do jiného micro:bitu i údaje o venkovní teplotě. Já se ovšem jako ukázkovou úlohu rozhodl připravit vnitřní teploměr, který hlídá teplotu, a v případě, že bude teplota moc dlouho vysoká, například pokud necháme moc dlouho zapnuté topení, nebo zase moc nízká, tak nás micro:bit hlasitým pípáním začne varovat. K sestavení této úlohy je potřeba senzor teploty a bzučák či reproduktor.

**Cíl a hodnocení:** Cílem této úlohy je chápat funkci senzoru teploty a jeho úspěšné zapojení a nastavení. Žáci zároveň pochopí zapojení bzučáku a jeho využití s bloky pro zvuk. Procvičí si své informatické myšlení využitím několika bloků „když“. Je dobré zde hodnotit vynalézavost žáků a jejich vlastní inovaci.

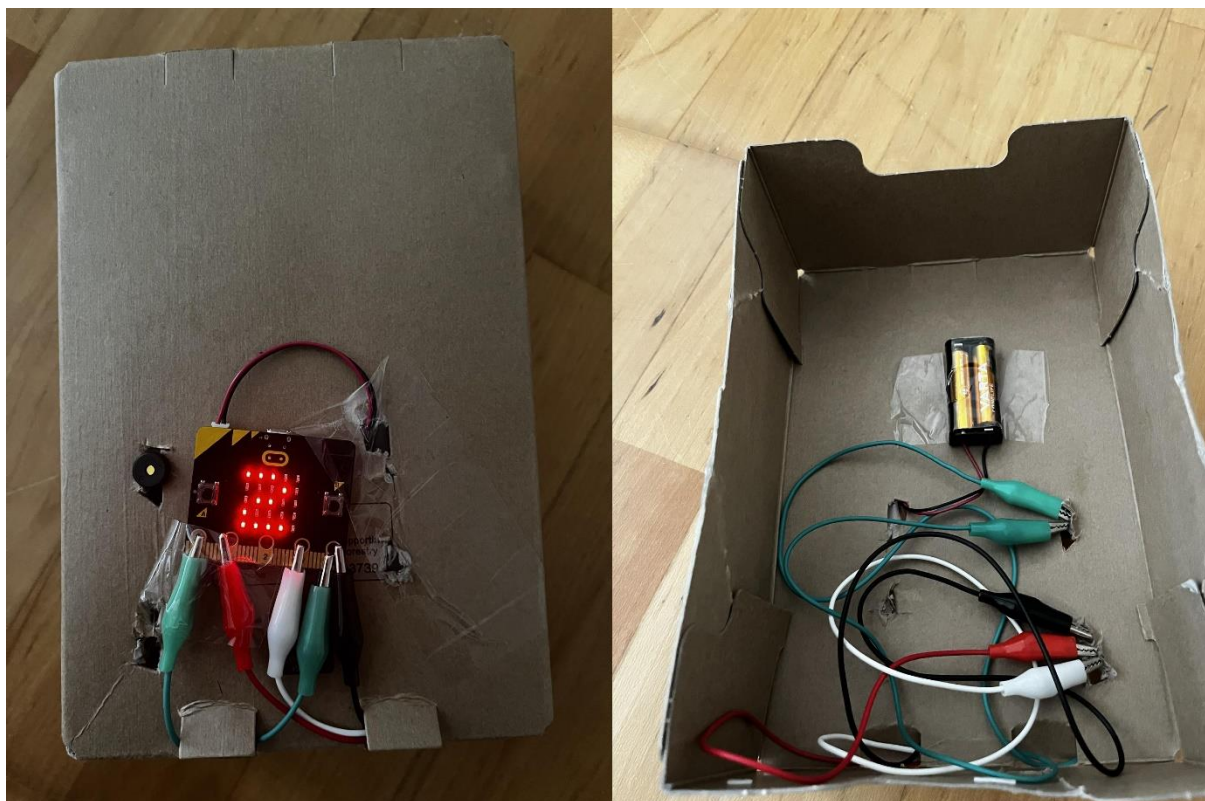


**Řešení pro teploměr:** V ukázkovém řešení jsem si vytvořil proměnnou „kriticky\_cas“, která měří, jak dlouho byl přesáhnut limit teploty. Každou vteřinu zobrazuji aktuální teplotu. Každou minutu se kontroluje, zda teplota přesáhla nastavený limit. V případě že ano, zvýší se proměnná o 1. Pokud bude teplota přesáжена po dobu, kterou si nastavíme, v mém případě 5 minut, spustí

se alarm, který nás varuje, že je na čase například vypnout topení. V případě, že teplota padne pod hranici, alarm se vypne a „kriticky\_cas“ se resetuje.

Žáci si zde mohou určit vlastní teploty, zvuk, který jim má hrát, nebo i dobu, po jakou musí být teplota přesažena. Je možné například přidat diody a alarm vytvořit i vizuálně.

**Zapojení a produkt:** Důležité je zde především správné zapojení samotného teploměru. Pokud je položen plochou částí nahoru, tak pin vlevo je „Vin“, který zapojíme na micro:bitu do 3 V. Prostřední pin je „Vout“, tento pin zapojíme do námi zvoleného pinu, například P1. Dále stačí zapojit bzučák, kde opět jeden pin půjde do námi zvoleného pinu, například P2, a druhý zapojíme do GND. K tvorbě výsledného produktu jsem využil krabici, do které jsem udělal díry tak, abych ní mohl provléct přípojky. Vzhledem k tomu, že chceme vidět teplotu, micro:bit musí být přilepen navrchu. Výsledný produkt tedy může vypadat například takto:





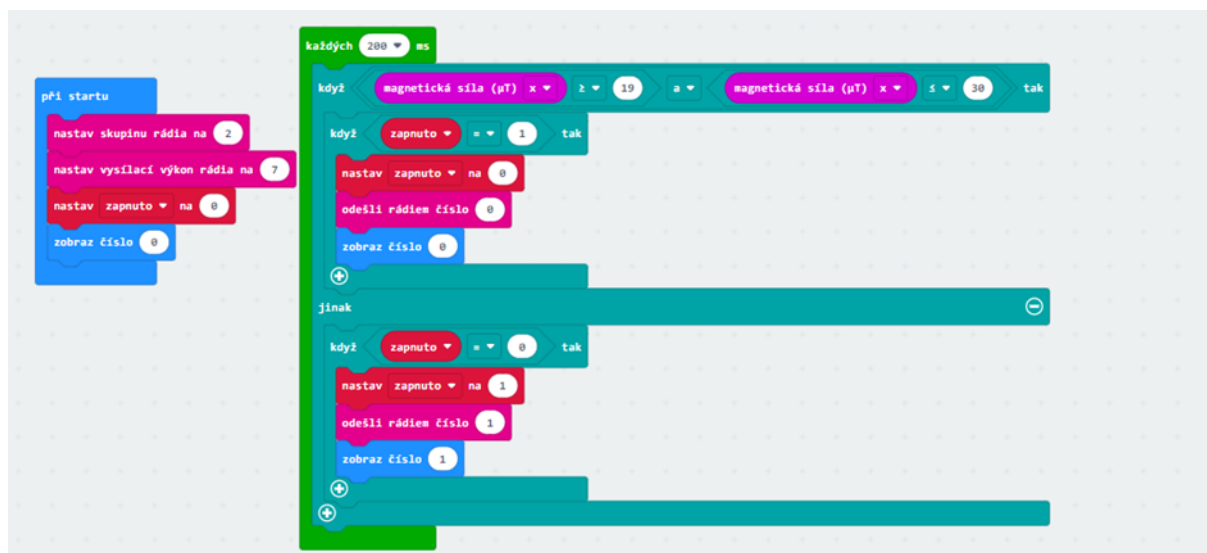
## Alarm na dveře

**Zadání:** Vytvořte dveřní alarm. Použijte 1-2 micro:bity a alespoň jednu periférii.

**Poznámka pro učitele:** Další možná úloha pro žáky je alarm. Tato úloha má velké množství řešení a zároveň může využít velké množství komponentů jako například dioda, bzučák, tlačítka. V ukázkovém řešení využívám bzučák. Micro:bit bude připevněn na dveře a v případě, že zaznamená otevření, vyšle do druhého micro:bitu zprávu, aby spustil alarm. Otevření je možné kontrolovat vícero způsoby, já využívám magnet, ale je možné využít i možnost měření pomocí kompasu nebo druhého micro:bitu.

**Cíle a hodnocení:** Tuto úlohu je možné řešit více způsoby. V každém případě žáci vymyslí způsob, jak detekovat, že se micro:bit nebo něco v okolí pohnulo. Pro kalibraci alarmu využijí pokročilejší funkci psaní do sériového portu. Zde můžeme hodnotit způsob jakým je toto detekováno. Žáci pochopí funkci rádia a vymyslí systém alarmu a jeho vypínání. Můžeme ocenit jejich inovaci a využití více periférií pro splnění tohoto úkolu.

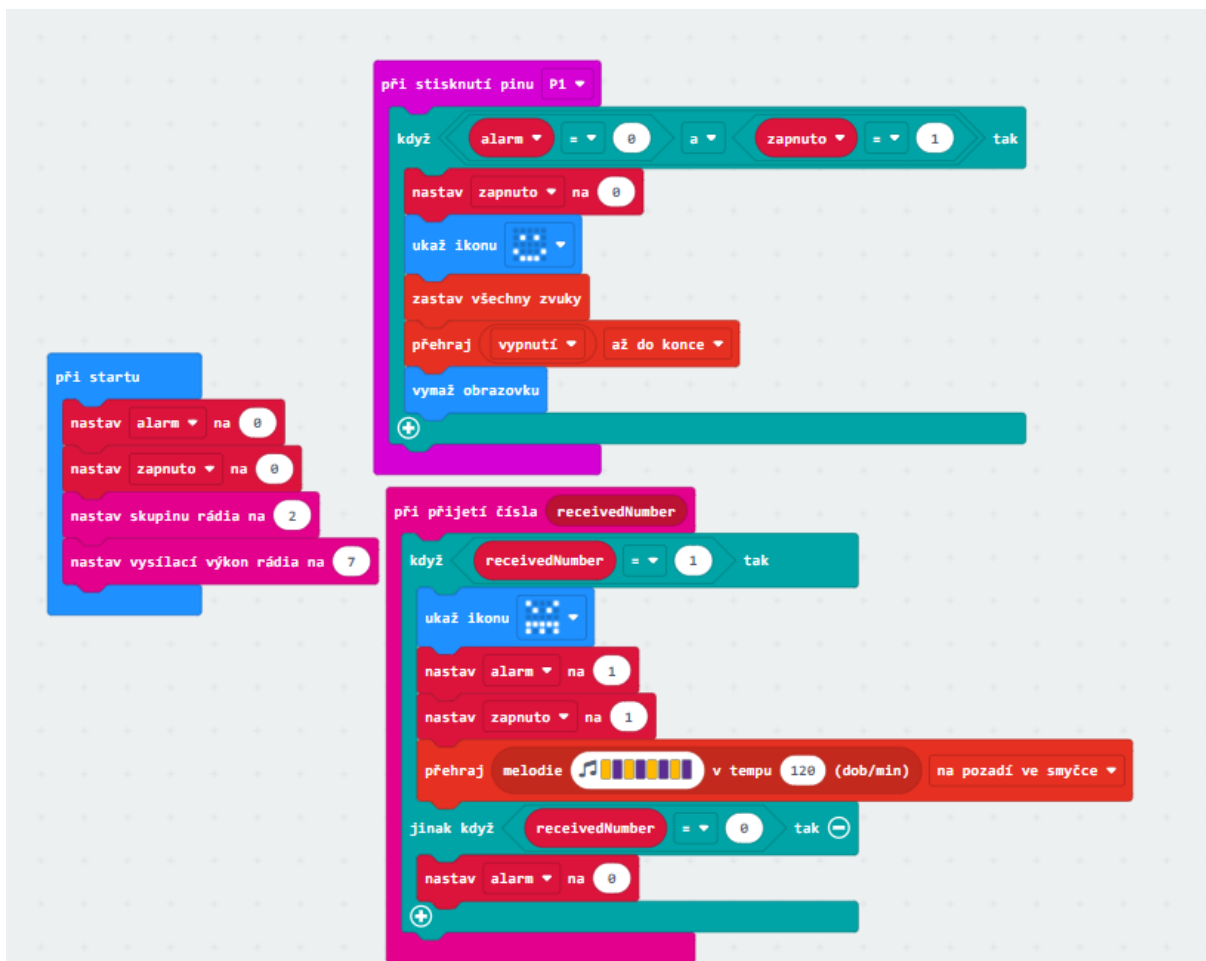
**Kód pro senzor otevření:** Pro kontrolu otevření využívám magnetu. Pro získání správných hodnot a nastavení, kdy jsou dveře považovány za zavřené, je vhodné využít pokročilé možnosti „sériová komunikace“, která nám do konzole zapisuje okamžitě naměřené hodnoty a která není zdržována vypisováním pomocí LED displeje. Po naměření potřebných hodnot a nastavení micro:bitu můžeme tuto funkci z kódu odstranit.



Toto je jedno z možných řešení pro senzor. Na začátku nastavím správnou skupinu a vytvořím si proměnnou „zapnuto“, která se stará o to, aby se neposílal stále stejný signál dokola. Každých 100ms micro:bit kontroluje, zda je jeho magnetické pole mezi nastavenými hodnotami, které značí zavřené dveře. V případě, že se dostane mimo tyto hodnoty, vyšle do

druhého micro:bitu signál alarmu. Pokud jsou zavřené na LED displeji se ukazuje 0, v opačném případě se ukazuje 1.

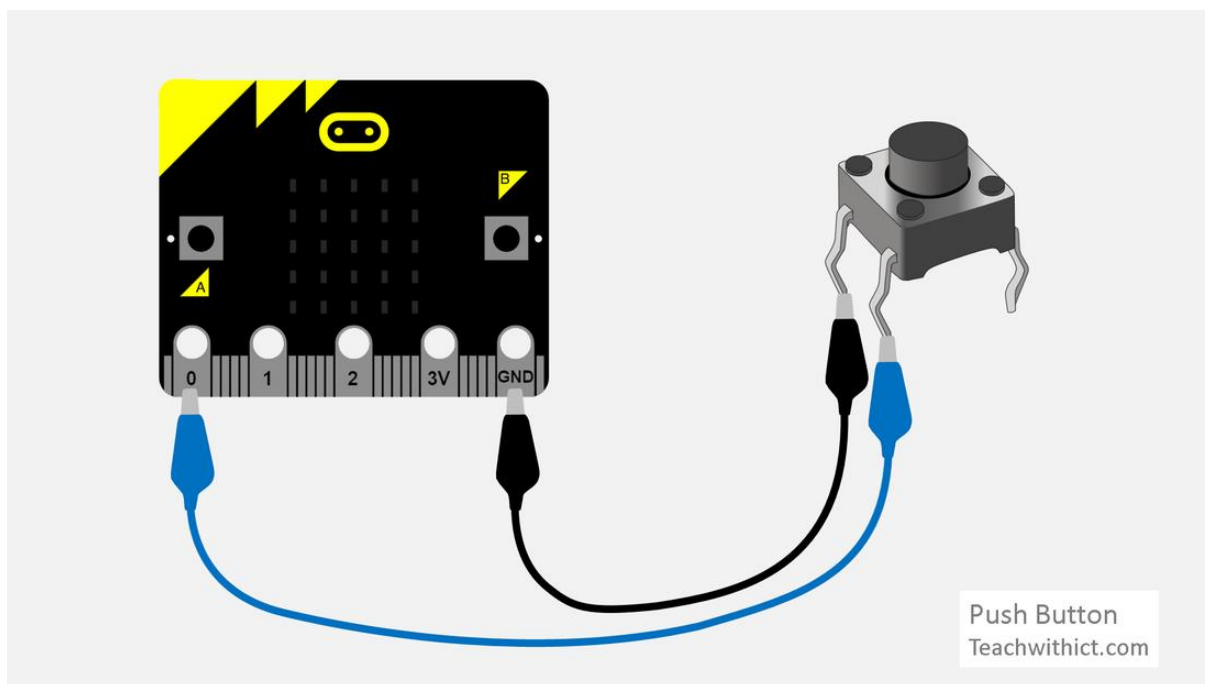
**Kód pro alarm:** Ke zvuku alarmu zde používám bzučák, je možné přidat například barevné diody. Po obdržení signálu se alarm spustí a je nutné ho manuálně vypnout. Žáci si zde mohou nastavit vlastní kombinaci tlačítek, pokud chtějí.



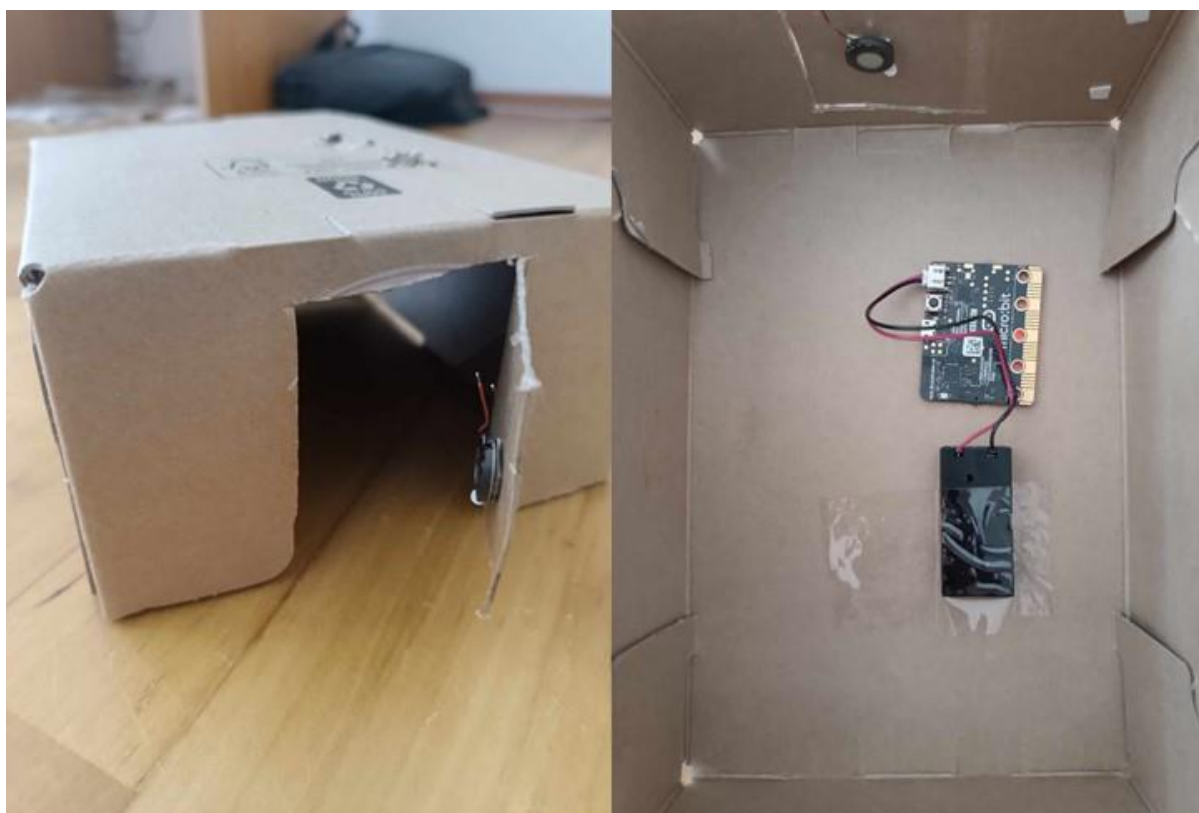
Mám zde 2 proměnné. Proměnná „alarm“ si ukládá poslední přijatý signál od senzoru. Proměnná „zapnuto“ kontroluje, zda je samotný alarm zapnutý. V případě, že přijde signál o spuštění alarmu, nastaví se tyto hodnoty na 1 a začne hrát melodie. Pokud chceme alarm vypnout je nutné, aby se dveře zavřely, tím se hodnota „alarm“ nastaví na 0 a poté musíme stisknout přidávané tlačítko. Poté se alarm vypne a znovu čeká na spuštění.

**Zapojení a produkt:** Na zapojení zde opět máme bzučák a tentokrát je k micro:bitu ještě přidávané tlačítko. Bzučák zapojíme tak, že první pin zapojíme k pinu, který si zvolíme na

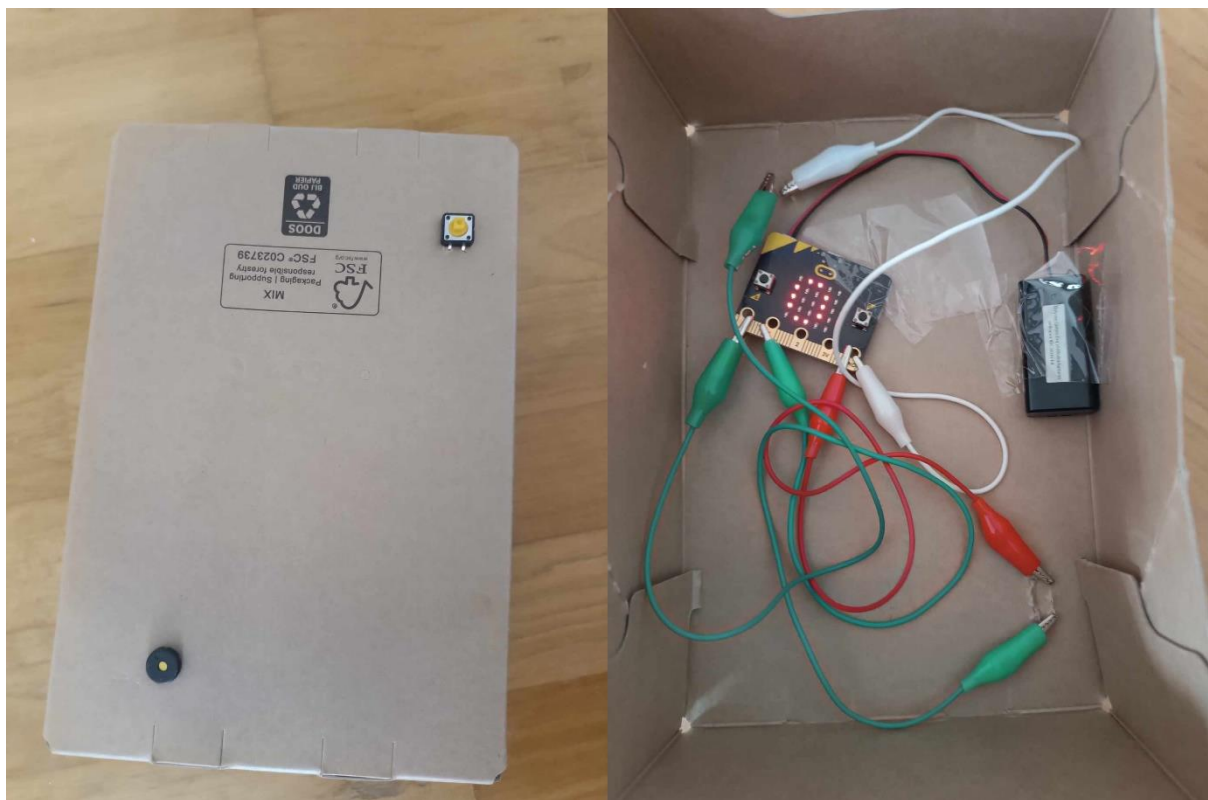
micro:bitu. Druhý půjde do části GND. Pro zapojení tlačítka opět využijeme jednoho zvoleného pinu, a druhý půjde do GND. Je důležité je zapojit stejně jako na přiloženém schématu:



Následně tento produkt můžeme vzít a opět využít nějaké kartonové krabice. Pro vyzkoušení si můžeme z jedné krabice vyříznout provizorní dveře, na které micro:bit přilepíme.



A pro samotný alarm si vzít druhou krabici, do které si uděláme díry pro tlačítko a bzučák a vše ostatní schováme dovnitř. Výsledný produkt může vypadat například takto:

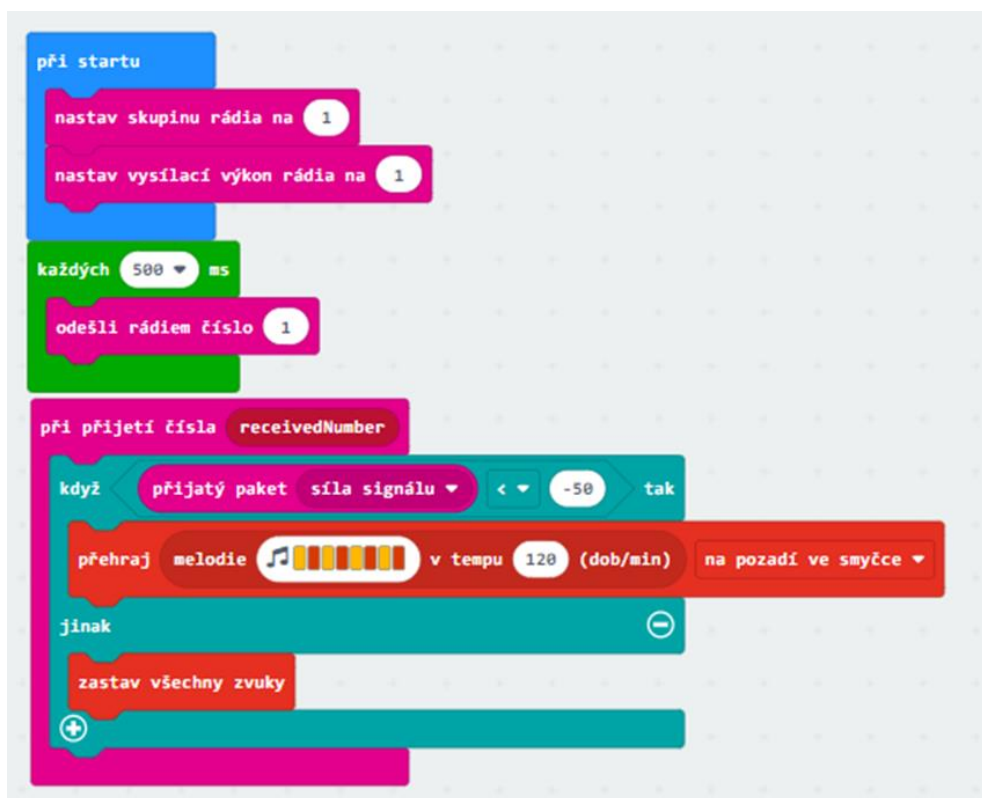


## Zabezpečení proti krádeži

**Zadání:** Vytvořte zařízení, která spustí alarm v případě, že se od sebe vzdálí. Využijte 2 micro:bity a alespoň jednu periférii.

**Poznámka pro učitele:** Další úloha je zařízení, které zabezpečuje proti krádeži nebo ztrátě. Tato úloha vyžaduje 2 micro:bity a bzučák. Tyto 2 micro:bity se od sebe nesmí vzdálit na delší vzdálenost, v případě, že mezi sebou ztratí kontakt, spustí alarm, který se nevypne do doby, dokud se k sobě opět nepřiblíží. Celý princip je založen na tom, že si micro:bity navzájem posílají signály a jakmile od sebe určený čas žádný signál neobdrží, spustí alarm. Na obou micro:bitech běží stejný kód, v teorii se dá takto propojit více zařízení.

**Cíle a hodnocení:** Žáci zde využijí pokročilejší funkce rádia, vymyslí způsob jakým hlídat vzdálenost mezi 2 micro:bity. Zároveň zde zapojí periférii, která dá o přílišném vzdálení vědět. Je zde důležité dát pozor na přesnost tohoto měření a to, zda micro:bity opět navážou spojení po přiblížení.



**Řešení pro zabezpečení proti krádeži** Zde je vidět, že používám proměnnou „přijatý paket síla signálu“. Tato proměnná mi měří, jak silný signál přišel od druhého micro:bitu, díky čemuž jsem schopen vnímat jejich vzdálenost. Proto spustím alarm v tu chvíli, co vzdálenost přesáhne nějakou hranici. Pomocí vysílacího výkonu si můžeme nastavit přesnost tohoto měření. Na

kratší vzdálenosti je lepší použít menší výkon, kdy je pak micro:bit schopen zaznamenat pohnutí každých pár centimetrů.

**Zapojení a produkt:** Zde je opět důležité především zapojení námi zvolené periferie, která nás bude varovat při vzdálení micro:bitů. Já zde opět zvolil bzučák, kde opět zapojíme jeden pin do pinu na micro:bitu. Pro samotný produkt jsem zvolil pouhou destičku vyříznutou z kartónové krabice vzhledem k tomu, že je u tohoto zařízení důležitá kompaktnost.

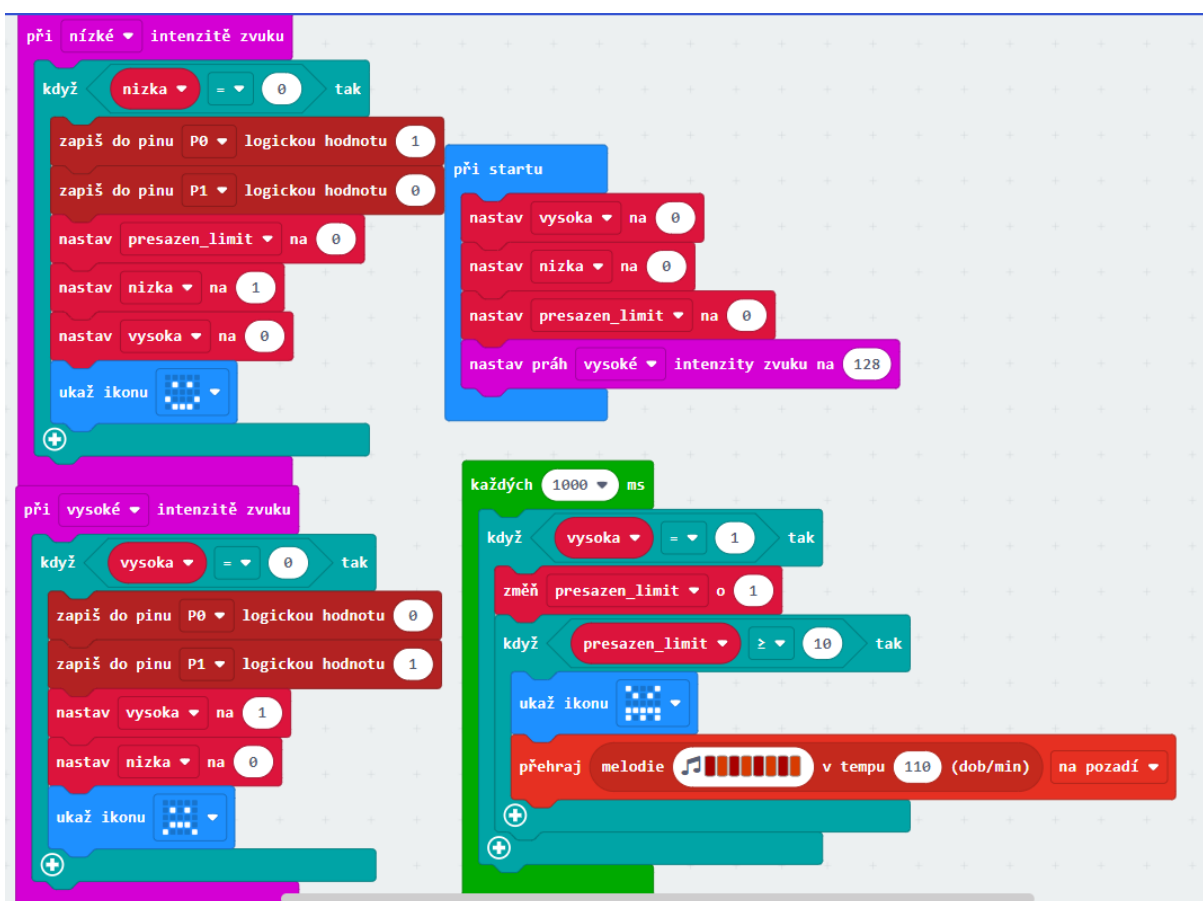
## Měřič hluku

Tato úloha využívá mikrofon, který je zakomponovaný v micro:bitu v2. Uvedl jsem jí zde jako ukázkou možného využití funkcí nové verze.

**Zadání:** Vytvořte měřič hluku, který vás varuje v případě přesažení hlasitosti. Využijte micro:bit v2 a diody.

**Poznámka pro učitele:** Tato úloha využívá nových bloků pro měření hluku. Měříme úroveň hluku v okolí a v závislosti na tom svítí dioda. Při přesažení kritické hranice může měřič spustit alarm, který nás varuje, že se máme ztišit, například v případě nočního klidu.

**Cíle a hodnocení:** Tato úloha žáky seznámí s novými funkcemi micro:bitu v2. Žáci využijí nově přidané bloky. Uvědomí si zde rozdíly mezi verzemi a limity micro:bitu v1. Je dobré si ověřit, zda chápou tyto rozdíly a jsou schopni využít nových funkcí.



**Řešení pro měřič hluku:** V tomto možném řešení využíváme bloky „při nízké/vysoké intenzitě zvuku“. V případě, že je intenzita zvuku nízká, svítí pouze jedna dioda. Pokud přesáhne danou hranici, první dioda zhasne a rozsvítí se druhá. V případě, že je hluk překročen delší dobu, tak micro:bit začne pípat pomocí bzučáku. Intenzitu zvuku si každý může nastavit, jak chce, stejně tak i varovný systém. Můžeme místo zvuku využít například toho, že budou diody blikat. Je



důležité zde upozornit, že bloky „při nízké/vysoké intenzitě zvuku“ se spustí pouze jednou při splnění události a poté až v případě, že intenzita opět změní z vysoké na nízkou či naopak. Proto zde musíme použít opakování, které průběžně kontroluje, jaký je stav. Zároveň máme u každého stavu nastavenou ikonu, která též ukazuje aktuální stav.

**Zapojení:** V tomto případě není nutné nic zapojovat, výhodou micro:bitu je, že má mikrofon i reproduktor již zabudovaný v sobě.