

ROBOTIKA PRO STŘEDNÍ ŠKOLY: PROGRAMUJEME MICRO:BIT POMOCÍ PYTHONU

Jiří Pech
Milan Novák



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MSMT
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



Poděkování

*Michal Kočer, Klára Pechová, Jan Wenig, Štěpán Bartoš, Karel Vlna, Vladimír Čížek,
Jana Kalová*



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

ROBOTIKA PRO STŘEDNÍ ŠKOLY: PROGRAMUJEME MICRO:BIT POMOCÍ PYTHONU

Mgr. Jiří Pech, Ph.D., PhDr. Milan Novák

Recenzent:
Mgr. Martin Cápay, PhD.

Vydavatel:
Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta

Obálka:
Mgr. Pavel Pfauser

Rok vydání: 2020



Podléhá licenci Creative Commons
Uveďte původ - Zachovejte licenci 4.0



9 788073 947873 >

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

METODICKÉ A PRACOVNÍ LISTY

V tomto souboru naleznete všechny potřebné metodické listy a pokyny pro učitele vztahující se k této učebnici. Pokud potřebujete další materiály navštivte následující webové stránky:

<https://github.com/jipech/PRIM-microbit>

Zde jsou navíc umístěny všechny zdrojové kódy a ukázkové prezentace k jednotlivým hodinám. Vše je zde i ve formátu pro Word nebo PowerPoint, takže každý učitel si může vše upravit, jak uzná za vhodné.

Struktura na webu je následující:

Každá kapitola učebnice na webových stránkách má čtyři nebo pět částí (adresářů):

Pro učitele – obsahuje kompletní text kapitoly včetně všech částí, návrhy výukových prezentací a průvodce hodinou s radami, jak vést výuku, seznamem potřebného materiálu, a odhad nutného času pro výuku. Ke všemu jsou k dispozici zdrojové kódy, učitel si vše může upravit dle svých potřeb.

Pro žáky – pracovní listy k jednotlivým hodinám. Až na výjimky se vejdou na jeden list papíru (oboustranně) a je možné je tak žákům vytisknout anebo dát k dispozici jako pdf soubor.

Samostudium – teoretický úvod k jednotlivým kapitolám, který opakuje a rozšiřuje probíranou látku a umožňuje žákům i učitelům hlouběji uchopit daná témata. Spojením těchto kapitol vznikl text nazvaný „učebnice“. Pokud by se např. zajímali rodiče o to, co děti probírají, je možné jim tento text rovněž doporučit.

Zdrojové kódy – všech řešených příkladů. Díky nim zejména rozsáhlejší programy není nutné opisovat.

Různé – fotografie, videa, obvody atd.

VÝPIS NA LED DISPLAY

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 nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

Časová náročnost

3 až 5 vyučovacích hodin po 45 minutách

PRŮVODCE HODINOU I-1

Studenti se seznámí s *micro:bit* a textovým editorem *Mu*. Naprogramují základní úlohu – běžící text „Ahoj světe!“.

Co bude v této hodině potřeba:

- PC s editorem *Mu*. Lze samozřejmě použít i jiný editor (viz kapitola Úvod) – pak je nutné upravit všechny pracovní listy, ne však zdrojové kódy příkladů. Můžete použít libovolný operační systém. Editor *Mu* stáhnete na stránce <https://codewith.mu/>
- *Micro:bit* s USB kabelem zakončeným *micro USB* koncovkou. Pozor nefungují všechny kabely. Pokud budete používat jiné než koupené spolu s *micro:bit*em, je nutné je předem vyzkoušet.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 10 minut

Vysvětlíte studentům, co se naučí v tomto předmětu:

- Poznájí výukovou platformu *micro:bit* (počítač, jednočip) a naučí se jí ovládat
- Naučí se některým principům elektronických obvodů
- Seznámí s jazykem Python ve verzi *MicroPython* anebo si jej připomenou

Rozdejte studentům pracovní listy a *micro:bit*y

2. krok 10 minut

Popište studentům *micro:bit*. Řekněte něco o jeho historii a možnostech a použití. Nechte studenty, aby si jej prohlédli. Promítejte během výkladu *micro:bit* na projektor z prezentace. Zmiňte jeho následující vlastnosti:

- matice 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 (bohužel v *MicroPythonu* nepoužitelné)
- možnost vzájemné komunikace
- 17 **GPIO** (general-purpose input/output) pinů, dále piny 3 V a GND. Tři piny GPIO (označené 0, 1, 2) a napájecí piny jsou větší pro snadné připojení vodičů s krokodýlky. Ostatní piny pro plnohodnotné použití vyžadují speciální základnu (shield).

3. krok 10 minut

Představte jazyk *Python* a *MicroPython* – zjednodušenou podmnožinu *Pythonu*, obsahující navíc knihovny pro práci s micro:bitem. Představte editor *Mu* a jeho možnosti. Při výkladu promítejte editor na projektor (např. z prezentace). Proberte krátce význam tlačítek.

- New, Load, Save – Nový program, nahrání programu z disku do editoru Mu, uložení (při prvním uložení se ptá na jméno programu)
- Flash – Přeložení programu do binárního kódu hex a jeho nahrání na připojený micro:bit
- Mode – Výběr módu editoru Mu. Je nutné mít nastaveno „BBC micro:bit“ (tato volba je defaultní)
- Files – zobrazí soubory na micro:bitu a soubory (programy) v domovské složce programu
- Repl – možnost zkoušet příkazy přímo na micro:bitu bez psaní programu, de facto příkazová řádka
- Zoom in, zoom out – velikost písma
- Theme – světlý text na tmavém pozadí a naopak
- Check – kontrola syntaxe. Většina chyb se projeví až při spuštění. Opětovné stisknutí vypne zobrazení chyb a následující jej opět zapne.
- Help – odkaz na stránky s nápovědou
- Quit – konec

4. krok 15 minut

Studenti napíší a spustí první program – `ahoj_sвете.py`. Řekněte studentům, aby si spustili editor Mu a připojili micro:bit USB kabelem. Do editoru napíší dva řádky kódu:

```
from microbit import *  
display.scroll("Ahoj světe")
```

Promítněte studentům kód na projektor a vysvětlete význam obou řádků.

Vysvětlete studentům, že **nelze používat české znaky** (a to ani v textech), protože je micro:bit neumí zobrazit.

Nechte studenty zkontrolovat syntaxi a řešte s nimi chyby.

První řádek budou možná již mít v editoru Mu přednastavený. Nechte studenty 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. Někdy, není-li micro:bit automaticky detekován, je nutné pro nahrání programu vybrat pro uložení micro:bit jako připojené externí zařízení.
- Microbit píše něco jiného – informaci o chybě. Můžete rovněž zkusit chybu nalézt stiskem klávesy check v editoru Mu.
- Program by měl končit odřádkováním a po něm následovat prázdný řádek

Pokud zbude čas, zkuste ještě program s nekonečnou smyčkou `ahoj_svete2.py`, jinak jej ponechte na začátek další hodiny.

PRŮVODCE HODINOU I-2

V této hodině nejprve rozšíříme příklad „Ahoj_sвете“ z minulé hodiny. Později přidáme dva další příklady (jeden z nich ve dvou modifikacích). Na těchto příkladech si současně připomeneme nebo se naučíme psaní cyklů v Pythonu.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 10 minut

Řešte úlohu s nekonečným výpisem textu „Ahoj světe“. Vysvětlíte studentům význam cyklu while True:

```
from microbit import *
while True:
    display.scroll("Ahoj světe")
    sleep(1000)
```

Obraťte pozornost na správnou syntaxi.

- True musí být s velkým T.
- Na konci řádku se zápisem cyklu musí být znak dvojtečky. Na ten se často zapomíná.
- Odsazení těla cyklu musí být o právě čtyři mezery. Editor Mu vám pomůže automatickým odsazením dalších řádků. Studenti mohou být z jiných verzí Pythonu například zvyklí na používání tabulátoru – to zde není možné.

2. krok – 25 minut

Napište dvěma různými způsoby program, který vypíše čísla od 1 do 10 a pak skončí.

Použijete postupně cykly for a while.

Zápis s cyklem for:

```
from microbit import *
for i in range(1, 11):
    display.scroll(i)
```

Vysvětlíte syntaxi programu. Jedná se o *cyklus s pevným počtem opakování*. Je třeba vysvětlit, že zápis range (1, 11) znamená v Pythonu rozsah od 1 do 10. Je to vždy o jednu méně, než je

mez vpravo – **častý zdroj chyb**. Dále je třeba říct, že za čárkou v závorce musí být mezera. Zkuste studentům říct některé další příklady zápisu range:

- `range(3, 1)` – cyklus se neprovede (3 je větší než 1)
- `range(1, 4, 2)` – cyklus se provede pro 1 a 3. Iterace je po dvou. Význam dosazuj do proměnné od jedné čísla zvětšená o 2 dokud je číslo menší než 4.
- `range(3, 1, -1)` – cyklus se provede pro 3 a 2. Sestupná iterace.

Nechte studenty případně vyzkoušet. Pokud to potřebujete, popis syntaxe cyklu `for` a rozsahu `range` je v příloze učebnice.

Zeptejte se studentů

Jaký je rozdíl mezi řetězcem (`string`) a celým číslem (`integer`)?

Řetězec je posloupnost znaků anglické abecedy, čísel a dalších znaků. Přesněji definováno, jedná se o znaky s ASCII kódem 32 až 127.

Definici celého čísla známe z matematiky. Zcela přesně v našem případě je to číslo bez desetinné čárky z intervalu `<-2147483648, 2147483647>`.

Zápis s cyklem `while`:

Jedná se o cyklus s neurčitým počtem opakování.

Pohovořte o dané syntaxi. Za příkazem `while` následuje podmínka a cyklus (odsazené řádky za dvojtečkou) se provádí tak dlouho dokud podmínka platí.

```
from microbit import *
i = 1
while (i < 11):
    display.scroll(i)
    i = i + 1
```

Všimněte si práce s proměnnými – definice proměnné na řádku 3 a změny její hodnoty na řádku 6. Pozor kolem znaku `=` a matematických operátorů **musí být mezery** – Mu je tam vyžaduje, ačkoliv v jiných verzích Pythonu nejsou nutné. Ukažte to studentům na kontrole syntaxe.

Pozor na mezery. Např. V zápisu: `i = i + 1` musí být mezera jak kolem znaku `=` tak okolo znaku `+`.

Zkuste se studentů zeptat, zda je jim bližší zápis s `for` nebo s `while`?

Vysvětlíte pojem negace podmínky – `not(podmínka)`

Je totéž jako v našem příkladu `while not(i > 11):` ?

Není správně by bylo: `while not(i >= 11):`

3. krok – 10 minut

Ukázka dalších funkcí pro objekt `display`:

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

Příklad zobrazí znak X pomocí příkazu `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 zůstane svítit tak dlouho, než zobrazíte něco jiného nebo smažete obrazovku pomocí `display.clear()`.

PRŮVODCE HODINOU I-3

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

Co bude v této hodině potřeba:

- PC se editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro studenty

1. krok 20 minut

Vyzkoušejte zobrazení předpřipravených obrázků. Micro:bit v této ukázce střídá náladu:

```
from microbit import *
display.show(Image.SAD)
sleep(1000)
display.show(Image.SMILE)
sleep(1000)
display.show(Image.HAPPY)
sleep(1000)
display.clear()
```

Zvažte, jakým způsobem studentům dát seznam obrázků. Zda jim poskytnout odkaz na web MicroPythonu nebo seznam, který je i v příloze této kapitoly, vytisknout.

Následuje další ukázka, která simuluje údery srdce:

```
from microbit import *
while True:
    display.show(Image.HEART)
    sleep(400)
    display.show(Image.HEART_SMALL)
    sleep(400)
```

Zeptejte se studentů:

Proč je použit příkaz sleep(400)?

Jeden běh cyklu takto trvá 0,8 sekundy, což vede na frekvenci 75 tepů za minutu.

2. krok 25 minut

Napište a vyzkoušejte následující program. Vzhledem k složitější syntaxi, náchylné k chybě při opisování, zvažte možnost studentům poskytnout zdrojový kód, aby vše nemuseli opisovat.

```
from microbit import *
raketa = Image("00900:"
               "05550:"
               "05550:"
               "09990:"
               "90909:")
display.show(raketa)
```

Pozor na syntaxi obrázku:

- Každý řádek kódu je řádek displeje
- Každý řádek je uvozen apostrofem a uvnitř končí dvojtečkou
- Čísla od 0 do 9 znamenají intenzitu světla (0 – nesvítí, 9 – svítí naplno)

Poskytněte studentům prostor pro sestavení vlastního obrázku.

Zápis Image je možný i na jeden řádek takto:

```
from microbit import *
raketa = Image("00900:05550:05550:09990:90909:")
display.show(raketa)
```

Důležitá webová adresa

Generátor obrázků:

<https://www.prf.jcu.cz/generator-led-matrix/index.htm>

Nutno nastavit matici 5x5 a jazyk Python

PRŮVODCE HODINOU I-4 (I-5)

Studenti se seznámí s pokročilejší grafikou na micro:bitu. Naučí se tvorbě animace a adresaci konkrétní diody displeje.

Jakým způsobem a zda vůbec učit tuto část ponecháváme na učitelích. Je možné tuto část vypustit buď zcela nebo první či druhou část. Nebo je naopak možné tuto kapitolu rozdělit do dvou samostatných hodin. Pokud učíte dvouhodinovky, je možné první část připojit k hodině III a ke druhé části v následující dvouhodinovce přidat opakování celé této části. Rozhodně by se s touto částí nejprve učitel měl dobře seznámit a rozhodnout se o způsobu výuky dle úrovně svých studentů.

První program v této kapitole je poměrně rozsáhlý. Zvažte proto možnost jeho zdrojový kód tentokrát žákům vhodným způsobem poskytnout, aby jej nemuseli opisovat. Pokud naopak je necháte kód opisovat, např. z důvodu procvičení ladění programu, pak počítejte s nutností rozdělit kapitolu do dvou hodin.

Z výše uvedených důvodů tentokrát neuvádíme časovou náročnost jednotlivých částí.

Co bude v této hodině potřeba:

- PC se editorem Mu.
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro studenty
- Volitelně: Připravený zdrojový kód programu z 1. kroku, umístěný tak, aby k němu měli žáci přístup.

1. krok

Napište nebo stáhněte následující kód a nahrajte jej do micro:bitu:

```
from microbit import *
raketa1 = Image("00900:"
                "05550:"
                "05550:"
                "09990:"
                "90909:")
raketa2 = Image("00900:"
                "05550:"
                "05550:"
                "09990:"
                "99999:")
raketa3 = Image("05550:"
                "05550:"
                "09990:"
                "99999:"
                "00000:")
raketa4 = Image("09990:"
                "99999:"
                "00000:"
                "00000:"
                "00000:")
raketa5 = Image("99999:"
                "00000:"
                "00000:"
                "00000:"
                "00000:")
raketa6 = Image("00000:"
                "00000:"
                "00000:"
                "00000:"
                "00000:")
raketa = [raketa1, raketa2, raketa3, raketa4, raketa5,
          raketa6]
display.show(raketa, delay=500)
```

Řekněte studentům:

- Jedná se vlastně o postupné zobrazení obrázků `raketa1` až `raketa6` po půl sekundě
- Struktura `raketa` se nazývá **list (seznam)** – jedná se o uspořádanou n-tici, u které záleží na pořadí a umožňuje opakovaný výskyt jednotlivých prvků

Úkol pro samostatnou práci:

Je možné vypustit obrázek `raketa6`? Pokud ano, je nutná úprava programu?

Ano je to možné, pokud místo toho přidáte k programu následující dva řádky:

```
sleep(500)
display.clear()
```

Máte-li čas, nechte studenty vytvořit vlastní animaci.

2. krok

Pro adresaci konkrétního bodu displeje slouží příkaz:

`display.set_pixel(X, Y, intenzita)` s následujícím významem:

- X – sloupec, zleva doprava od 0 do 4
- Y – řádek shora dolů od 0 do 4
- intenzita – jas diody, 0 vypnutá, 9 zapnutá naplno.

Bod vlevo nahoře má souřadnice 0, 0, a bod vpravo dole 4, 4.

V následujícím programu je použit **generátor náhodných čísel**. Pro jeho použití je nutno zavést knihovnu `import random` a pak je možné použít funkce `random.randint(A, B)`, která vrací náhodné číslo z uzavřeného intervalu od A do B.

```
from microbit import *
import random
while True:
    x = random.randint(0, 4)
    y = random.randint(0, 4)
    intenzita = random.randint(0, 9)
    display.set_pixel(x, y, intenzita)
    sleep(10)
```

Program v nekonečném cyklu načítá náhodné souřadnice a intenzitu a s danými parametry rozsvěcí diodu, celkový dojem trochu připomíná hvězdy na noční obloze. Časová prodleva je přidána, aby nedocházelo k příliš rychlému blikání.

Zeptejte se studentů:

Jak pracuje generátor náhodných čísel?

Náhodné číslo je generováno např. na základě času od zapnutí a teploty okolního prostoru.

Tato dvě čísla se mohou sečíst a dělit nějakým prvočíslem a pak vzít číslo na konkrétní pozici jako výsledek.

Jedná se o analogové či o digitální zobrazení?

Jedná se o diskretizaci analogového zobrazení – výsledek může nabývat více než dvou hodnot, ale omezený počet (10).

Program nyní upravte:

```
from microbit import *
import random
while True:
    x = random.randint(0, 4)
    y = random.randint(0, 4)
    if (display.get_pixel(x, y)):
        display.set_pixel(x, y, 0)
    else:
        display.set_pixel(x, y, 9)
    sleep(10)
```

Řekněte studentům:

- Jedná se de facto o digitalizaci, neboť v této úloze jsou použity pouze dvě úrovně rozsvícení diody 0 a 9.
- Funkce `display.get_pixel()` zjistí aktuální úroveň světla diody na dané souřadnici. Pokud vrátí hodnotu 0, podmínka není splněna. Následně je nastavena opačná intenzita svícení, než byla před tím.
- Pozor na dvojí úroveň odsazení. Ve druhé úrovni musí být 8 znaků (násobek 4).

2 PRÁCE S TLAČÍTKY

Co se naučíte

- Ovládat obě programovatelná tlačítka
- Psát programy reagující na stisk tlačítka
- Význam logických spojek and a or
- Vnořené funkce

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

Časová náročnost

Jedna vyučovací hodina 45 minut

PRŮVODCE HODINOU II-1

Co bude v této hodině potřeba:

- PC se editorem Mu.
- Micro:bit s USB kabelem zakončeným micro USB.
- Pokud je k dispozici, tak dataprojektor.
- Prezentaci k této lekci.
- Pracovní listy pro studenty.

1. krok 5 minut

Vysvětlíte, že micro:bit má celkem tři tlačítka. Dvě na přední straně a jedno na zadní straně.

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

Na přední straně pak má dvě programovatelná tlačítka označená A a B. V MicroPythonu pro ně existují dvě proměnné `button_a` a `button_b`. Pokud studenti vědí, co je to objektové programování, použijte správný pojem objekty `button_a` a `button_b`. Můžete rovněž využít tuto látku k prvnímu seznámení s objektovým programováním.

2. krok 10 minut

Začněte jednoduchým příkladem:

```
from microbit import *
while True:
    if button_a.is_pressed():
        display.show(Image.HAPPY)
    if button_b.is_pressed():
        display.show(Image.SAD)
    sleep(100)
    display.clear()
```

Příkazy `if_button_X_is_pressed` slouží pro dotaz, zda je konkrétní tlačítko stisknuto. Nechte studenty přijít na jejich význam. Funkce (metoda) vrací hodnotu 1 (True, Ano – stisknuto) nebo 0 (False, Ne – nestisknuto). Existuje i funkce `button_a.was_pressed()`, která testuje, zda tlačítko bylo stisknuto od minulého testování nebo od začátku programu (pokud dosud k žádnému testování nedošlo).

Pozor na správné odsazení bloku ve druhé úrovni (po `if`) – musí být 8 mezer.

3. krok 15 minut

Nyní si vysvětlíme význam logických spojek **and** (a) a **or** (nebo). Začněte tímto příkladem:

```
from microbit import *
while True:
    if (button_a.is_pressed()) and (button_b.is_pressed()):
        display.show(Image.HEART)
    sleep(100)
    display.clear()
```

Spojka `and` mezi dvěma testy na řádku 3 má význam `a` – obě podmínky musí být splněny současně.

V programu změňte `and` za `or`:

```
from microbit import *
while True:
    if (button_a.is_pressed()) or (button_b.is_pressed()):
        display.show(Image.HEART)
    sleep(100)
    display.clear()
```

Spojka `or` má význam `nebo`. Stačí když je splněna alespoň jedna z podmínek.

4. krok 15 minut

Funkce `get_presses()` vrací počet stisknutí daného tlačítka od startu nebo poslední kontroly. Napište a spusťte následující kód:

```
from microbit import *
sleep(10000)
display.show(button_a.get_presses())
```

Neřešený závěrečný příklad: Nechte studenty naprogramovat postřehovou hru. Na Micro:bitu se bude střídavě náhodně zobrazovat A nebo B a hráč bude muset do určité doby stisknout odpovídající tlačítko. Hra může například skončit stiskem obou kláves současně anebo může mít pevný počet pokusů. Doba zobrazení a čekání na stisk může být konstantní nebo se může snižovat dle počtu úspěšných pokusů. Na závěr může být vyhodnocení např. Procentem úspěšných pokusů. Pro volbu A nebo B použijte generátor náhodných možností z kapitoly 1.

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

Řekněte studentům ať si přinesou na příští hodinu sluchátka nebo repráček s jackem. Případně si připravte dostatečný počet piezobuzzerů.

Doporučujeme, aby studenti měli sluchátka, ať se vzájemně nepřehlušují ráumusem. Vy si naopak připravte repráčky, ať vše můžete dobře demonstrovat.

POZNÁMKA – máte-li nový **micro:bit V2** nepotřebujete externí zdroj zvuku, tento micro:bit již má buzzer pevně připojený.

3 HUDBA

Co se naučíte

- Připojit k micro:bitu reproduktor, buzzer nebo sluchátka
- Přehrát přednastavený zvuk
- Naučíte micro:bit mluvit
- Vytvořit vlastní melodii

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reproduktor nebo sluchátka s jackem, popřípadě piezzo buzzer

Časová náročnost

Dvě až čtyři (v případě samostatné práce) vyučovací hodiny pro 45 minutách.

Microbit V2 vydaný v závěru roku 2020 již obsahuje piezzo buzzer a hardwarové konstrukce popsané v těchto materiálech tak již nejsou třeba. Všechny zde uvedené programy nadále fungují beze změny.

PRŮVODCE HODINOU III-1

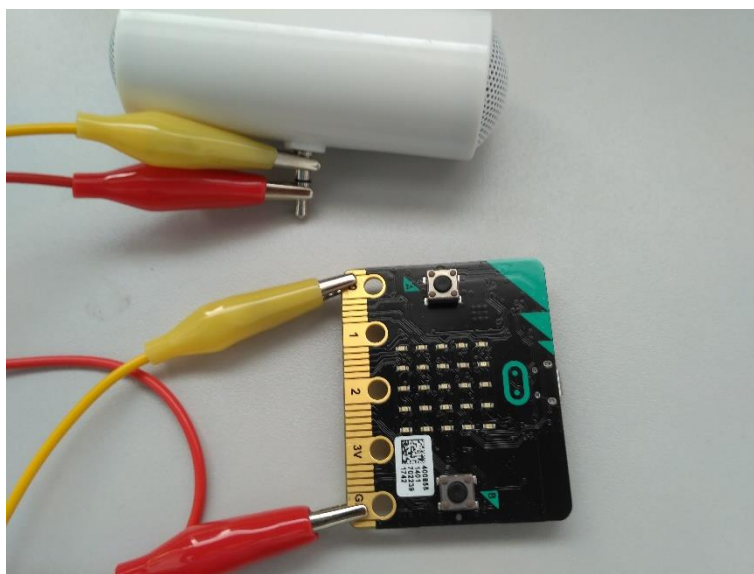
Studenti si připojí k micro:bitu hardware pro přehrání zvuku a ozvučí tak své projekty.

Co bude v této hodině potřeba:

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

1. krok 15 minut

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



Micro:bit V1 nemá přímý audio výstup, ale připojení externího reproduktoru je velmi snadné. Budete potřebovat dva vodiče nejlépe opatřené na koncích krokodýlky. Ty na dolní straně micro:bitu připněte jeden na výstup GND a druhý na výstup 0. Na základě osobních zkušeností můžeme potvrdit, že druhý vodič může být i na 1 anebo na 2 a dokonce pak může být zvuk i lepší. Záleží na použitém zařízení. První vodič musí každopádně být na GND. Druhý konec vodičů připojte na jack 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, ale není to pravidlem, možná budete muset trochu experimentovat. Také můžete použít jako výstup piezzo buzzer, pak prostě připojíte každý vodič k jednomu z pinů. Viz následující obrázek.

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

2. krok 15 minut

Vyzkoušejte přehrávání na připravené melodii. Zapište následující kód, odlad'te a nahrajte do micro:bitu:

```
from microbit import *  
import music  
music.play(music.NYAN)
```

Na řádku 2 je informace o přidání knihovny pro přehrání zvuku. Na řádku 3 je příkaz pro přehrání přednastavené melodie. Tento zvuk je poměrně dlouhý a poskytuje tak čas nastavit správné připojení výstupu. Pokud nic neslyšíte a myslíte, že je vše v pořádku a melodie již skončila, pak zmáčkněte na micro:bitu tlačítko reset.

Seznam připravených melodií je na konci této kapitoly. Podobně jako u přednastavených obrázků jej vhodným způsobem poskytněte studentům.

Odlad'te u všech studentů přehrávání hudby. Pokud někomu hudba nehraje, zkuste jiný hardware. Tato melodie je vhodná pro testování – je dlouhá a výrazná.

3. krok 15 minut

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

```
from microbit import *  
import music  
while True:  
    if button_a.is_pressed():  
        display.show(Image.HAPPY)  
        music.play(music.POWER_UP)  
    if button_b.is_pressed():  
        display.show(Image.SAD)  
        music.play(music.POWER_DOWN)  
    display.clear()
```

Tomuto příkladu by již žáci měli rozumět. Ověřte.

Pokud zbývá čas, nechte studenty upravit předchozí příklad dle nálady.

PRŮVODCE HODINOU III-2

Studenti si na micro:bitu připraví vlastní melodii a naučí jej mluvit.

Co bude v této hodině potřeba:

- PC s editorem Mu
- 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 studenty

1. krok 10 minut

Rozdejte studentům micro:bity a kabely. Nechte je připojit sluchátka. Napište a odlaďte následující program:

```
from microbit import *  
import speech  
speech.say("Hello", speed = 100)
```

Na řádce 2 se zavádí knihovna pro hovor a na řádce 3 je zadán příkaz pro mluvení. Zde micro:bit pozdraví. Parametr `speed=100` je nepovinný a je možné jej včetně čárky vynechat. Defaultní hodnota je 72, ale přijde nám, že při této hodnotě mluví micro:bit příliš rychle. Čím vyšší číslo, tím je řeč pomalejší a naopak.

Pozor micro:bit mluví pouze anglicky a je tedy nutné použít anglickou transkripci. Např. „Josef“ je třeba napsat jako „Yoseph“ atd. A pozor stejně jako při výstupu na displej nepoužívejte české znaky.

Dokumentace doporučuje zapojit pro hovor sluchátka (repráky) mezi porty 0 a 1 (a ne 0 a GND jako u hudby). A skutečně zvuk je v tomto případě silnější a čistší než mezi 0 a GND. Řekněte to žákům. Nezapomeňte na další části vodiče navrátit mezi 0 a GND.

Řekněte studentům, ať zkusí naučit micro:bit říkat jejich jméno a příjmení (bez háčeků a čárek).

2. krok 20 minut

Nechte studenty napsat a odladit následující program, který přehraje melodii ovčáci čtveráci. V tomto případě je možné pro zmenšení počtu chyb tento program vhodným způsobem studentům nasdílet. Melodie je poměrně primitivní, pokud máte mezi studenty hudebníky, určitě jí upraví:

```
from microbit import *
import music
nota = ["C4:4", "R:1", "E4:4", "R:1", "G4:4", "R:4", "C4:4",
        "R:1", "E4:4", "R:1", "G4:4", "R:4",
        "E4:2", "R:1", "E4:2", "R:1", "D4:2", "R:1", "E4:2",
        "R:2", "F4:2", "R:1", "D4:2", "R:1",
        "E4:2", "R:1", "E4:2", "R:1", "D4:2", "R:1", "E4:2",
        "R:2", "F4:2", "R:1", "D4:2", "R:1",
        "E4:4", "R:1", "D4:4", "R:1", "C4:4"]
music.play(nota)
```

Datová struktura `nota` je **seznam**, který by již měli studenti znát z kapitoly o animaci. Zkuste se jich zeptat.

Význam jednotlivých tónů je: C4:4 znamená nota C ve čtvrté oktávě (0 – nejnižší, 8 – nejvyšší) o délce 4. Nota R znamená pauzu (rest). Příkaz `music.play(nota)` pak daný záznam přehraje.

3. krok 15 minut

Vyzvěte studenty ať si sestaví vlastní melodii nebo ať naprogramují přehrání nějaké známé melodie.

Doporučení

Touto hodinou končí úvodní část seznamování s micro:bitem. Nyní se nabízí možnost zadání nějaké samostatné nebo týmové práce.

Navrhujeme, abyste nyní studentům zadali po dvojicích (nebo i větších skupinách) následující úlohu: Vytvořte pomocí dvou nebo tří micro:bitů animaci s melodií. Jeden micro:bit se bude starat o animaci a druhý k tomu bude hrát melodii. Popřípadě na třetím micro:bitu může probíhat nějaký hovor. Upozorněte studenty, že je třeba se nějak synchronizovat, např. současně stisknout tlačítka na všech micro:bitech. Později se studenti naučí též synchronizaci pomocí rádia, která by byla vhodnější, ale zatím jí nebudou znát.

4 POLOHA

Tato kapitola se věnuje orientaci micro:bitu a jeho pohybu v prostoru

Co se naučíte

- Pracovat s vestavěným akcelerometrem, pochopit jeho možnosti a využít jej
- Používat gesta
- Pracovat s vestavěným kompasem
- Používat detekci magnetického pole

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukční nebo sluchátka s jackem, popřípadě piezzo buzzer

Časová náročnost

Čtyři vyučovací hodiny po 45 minutách

Microbit V2 vydaný v závěru roku 2020 již obsahuje piezzo buzzer a hardwarové konstrukce (připojování sluchátek) popsané v těchto materiálech tak již nejsou třeba. Všechny zde uvedené programy nadále fungují beze změny.

PRŮVODCE HODINOU IV-1

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

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukční nebo sluchátka s jackem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

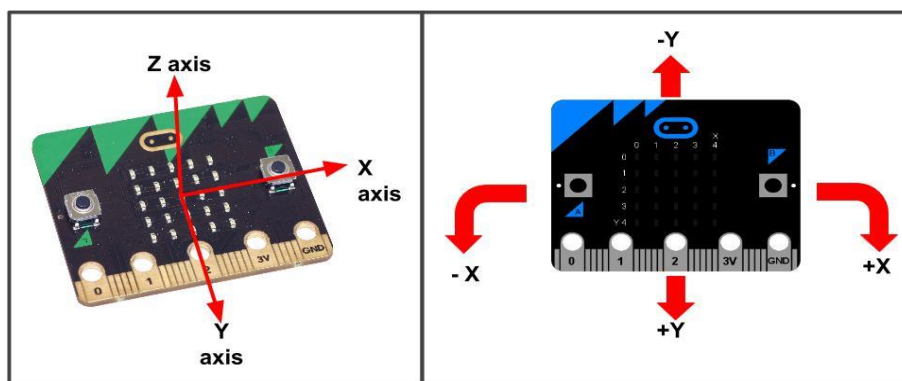
Rozdejte studentům micro:bity a kabely. Řekněte jim ať si připraví sluchátka. Raději mějte připravená sluchátka pro ty, kteří si je zapomenou.

Programy v této lekci jsou "dlouhé". Doporučujeme poskytnout studentům zdrojový kód a tak vše urychlit.

Vysvětlíte studentům pojem akcelerometr – zařízení pro zjištění aktuální orientace, směru pohybu, zrychlení, popř. volného pádu.

Zeptejte se studentů na využití akcelerometru – herní ovladač, generátor událostí. Program reagující na zatřesení, změnu orientace.

Vysvětlíte studentům, jak fungují osy x , y a z u micro:bitu. Je-li položen vodorovně piny k vám, je osa x rovnoběžně s vámi (a sleduje tedy náklon vlevo a vpravo), osa y pak směrem k vám a sleduje náklon od vás k vám. Osa z v tomto případě není sledovatelná. Použijete ji pouze, pokud je micro:bit „postaven“ na konektorech a naklání se k vám a od vás.



Poloha os a význam plus a minus ve směru os.

Zdroj <http://fibasile.fabcloud.io/microbit-notebook/accelerometer/>

Zapište a odlaďte následující program ukazující pohyb ve směru osy x (vlevo a vpravo):

```
from microbit import *
mez = 400
while True:
    naklon = accelerometer.get_x()
    if naklon > mez:
        display.show("P")
    elif naklon < -mez:
        display.show("L")
    else:
        display.show("-")
```

Tento program ukazuje princip sledování pohybu micro:bitu ve směru osy x (náklon vlevo a vpravo). Proměnná mez určuje hodnotu, od které považujeme micro:bit za nakloněný vpravo či vlevo.

Nechte studenty sledovat, jak program funguje vzhledem k pozici micro:bitu v prostoru. Nechte je měnit hodnotu proměnné mez. Poté změňte osy. Zcela vždy postačí nahradit `get_x` za `get_y` nebo `get_z`. Toto je vlastně způsob, jak můžete micro:bit použít jako dálkový ovladač, až se naučíte v některé z příštích hodin, přenášet data mezi micro:bity.

2. krok 20 minut

Nyní si postavíte jednoduchý simulátor thereminu. Theremin je nástroj, který ovládáme pohybem rukou bez dotyku nástroje. Pohybem jedné ruky určujeme výšku tónu a pohybem druhé, pak jeho délku.

Zapište a odlaďte následující program:

```
from microbit import *
import music
while True:
    x = accelerometer.get_x()
    y = accelerometer.get_y()
    if (x < -1000):
        ton = "C4"
    elif (x < -700):
        ton = "D4"
    elif (x < -400):
        ton = "E4"
    elif (x < -100):
        ton = "F4"
    elif (x < 200):
        ton = "G4"
    elif (x < 500):
        ton = "A4"
    elif (x < 800):
        ton = "B4"
    else:
        ton = "C5"
    if (y < -500):
        nota = ton
    elif (y < 0):
        nota = ton + ":2"
    elif (y < 500):
        nota = ton + ":4"
    else:
        nota = ton + ":8"
    music.play(nota)
```

Připojte k micro:bitu repráčky (sluchátka), dle postupu z minulých lekcí (mezi piny 0 a GND). Natáčením micro:bitu vpravo a vlevo regulujete výšku tónu, dopředu dozadu jeho délku. Ponechte studenty upravovat rozsah tónů, citlivost atd.

PRŮVODCE HODINOU IV-2

V této hodině se studenti naučí pracovat s gesty.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok – 5 minut

Vysvětlíte studentům, co jsou to gesta – nějaké konkrétní pohyby micro:bitu, např. otočení, zatřesení, volný pád atd. Jejich úplný seznam včetně jmen je v příloze k této kapitole. Zeptejte se studentů, jak by jednotlivá gesta mohli využít k různým činnostem.

2. krok – 15 minut

Napište a odlaďte následující program:

```
from microbit import *
while True:
    gesture = accelerometer.current_gesture()
    if gesture == "face up":
        display.show(Image.HAPPY)
    else:
        display.show(Image.ANGRY)
```

Tento program zobrazuje smajlík dle orientace micro:bitu v prostoru. Pouze je-li micro:bit obrácen displejem vzhůru a je v klidu, zobrazuje se úsměv, jinak nakloníme-li jej na libovolnou stranu, zobrazí se zamračená tvář.

Nechte studenty vyzkoušet i jiná gesta a smajlíky. Pokud budou testovat volný pád, měli by si dát pozor na rozbití micro:bitu. Poradte jim, aby v případě zobrazení smajlíku např. při volném pádu, přidali pauzu např. tři sekundy, aby byl smajlík chvíli zobrazen.

3. krok – 25 minut

Nechte studenty zapsat a odlažit následující program. Jedná se o variaci hry Magic 8 ball. Micro:bit v případě zatřesení na displeji zobrazuje náhodné odpovědi. Řekněte studentům, že se mají v duchu soustředit na nějaký problém, pak požádat micro:bit o jeho řešení a zatřást s ním.

Program je delší – zvažte následující možnosti. Program studentům nahrajte na místo, kde k němu budou mít přístup anebo jej připravte pouze s možnostmi ano, ne, možná jako je to na prezentaci a nechte studenty doplnit další možnosti. Nezapomeňte připomenout, že nelze používat háčky a čárky.

Nechte studenty libovolně upravovat možnosti.

```

from microbit import *
import random
odpovedi = [
    "Jiste",
    "Urcite",
    "Bez obav",
    "Ano, ano, ano",
    "Uvidime, uvidime",
    "Pravdepodobne",
    "To vypada dobre",
    "Ano",
    "Zeptej se pozdeji",
    "Ted nevim",
    "Nelze urcit",
    "Radeji ne",
    "Me vnitřní já říká ne",
    "Urcite ne",
    "Ne",
    "Nikdy"]
while True:
    display.show('8')
    if (accelerometer.was_gesture("shake")):
        display.clear()
        sleep(1000)
        display.scroll(random.choice(odpovedi))
        sleep(10)

```

Aby micro:bit neodpovídal pořád dokola je nutné použít funkci `accelerometer.was_gesture()`, která bere pouze gesta od posledního volání.

PRŮVODCE HODINOU IV-3

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

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

Kroky

Látku této hodiny pravděpodobně nestihnete. V takovémto případě klidně zbytek odložte na další hodinu, která je naopak krátká. Z tohoto důvodu není ani tato (a příští) hodina dělena na jednotlivé kroky.

Proberte se studenty pojmy kompas a azimut. Zeptejte se, zda těmto pojmům rozumí.

Micro:bit obsahuje integrovaný 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. Základní použití si můžete ukázat na následujícím programu:

```
from microbit import *
compass.calibrate()
while True:
    display.scroll(compass.heading())
    sleep(1000)
```

Pro kalibraci je nutno otáčet micro:bitem tak dlouho, než displej zaplníme svítícími diodami. Na micro:bitu vám vždy před kalibrací proběhne instrukce, jak postupovat. Po zaplnění displeje je třeba několik vteřin (cca. 5) počkat, než se na displeji objeví smajlík. Micro:bit položte na rovnou plochu nebo jej držte co nejvíce rovně. Micro:bit nyní ukáže na displeji *azimut*. Směr azimutu je přímo od displeje nahoru.

Tuto kalibraci je nutné provést před každým použitím kompasu. Vysvětlete to studentům, ať nejsou překvapeni.

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

Vyzkoušejte si rovněž co se stane, když kolem micro:bitu pohybujete magnetem nebo zmagnetizovaným předmětem (nůžky, šroubovák ...).

Nyní program upravte tak, aby micro:bit ukazoval symboly světových stran S, V, J, Z. Za sever budeme považovat intervaly úhlů <0,45> a <316, 359>, za východ <46, 135>, za jih <136, 225> a za západ <226, 315>.

```
from microbit import *
compass.calibrate()
while True:
    uhel = compass.heading()
    if (uhel < 46):
        display.show("S")
    elif (uhel < 136):
        display.show("V")
    elif (uhel < 226):
        display.show("J")
    elif (uhel < 316):
        display.show("Z")
    else:
        display.show("S")
    sleep(1000)
```

Program nyní upravte tak aby ukazoval na displeji micro:bitu směr na sever. Využijte při tom obrázek Image.ALL_CLOCKS. Jedná se vlastně o pole dvanácti obrázků, které se volají Image.ALL_CLOCKS[uhel], kde uhel je číslo od 0 do jedenácti. Na displeji pak ukazují čáru (lépe křivku) od středu micro:bitu ve směru malé hodinové ručičky pro hodinu o hodnotě proměnné uhel. Pozor namísto hodnoty 12 směr nahoru ukazuje hodnota 0. Můžete si to ověřit následujícím programkem:

```
from microbit import *
for uhel in range(0, 12):
    display.show(Image.ALL_CLOCKS[uhel])
    sleep(1000)
    display.clear()
```

Program, který ukazuje směr sever je v následujícím výpisu:

```
from microbit import *
compass.calibrate()
while True:
    uhel = ((compass.heading()-15) // 30)
    display.show(Image.ALL_CLOCKS[uhel])
```


Pokud věříte tomu, že to vaši studenti pochopí, vysvětlete jim výpočet na čtvrtém řádku. Je vysvětlen v teoretickém základu. Jinak pouze studentům řekněte, že se jedná o výpočet severu z daného azimutu.

Proberte se studenty význam operací // (dělení beze zbytku) a % (zbytek po dělení).

Otázky (nejen) pro zvědavé:

- Lze nahradit číslo 15, číslem 375? Jak musíme program upravit?
- Můžeme místo % 12 napsat + 12? Jak musíme program upravit?

Úkoly:

- Upravte program pro zobrazení azimutu pomocí šipek Image.ARROW_N, Image.ARROW_NE atd.
- Upravte program pro zobrazení azimutu tak, aby zobrazoval astronomický azimut 0 je na jihu, 90 západ, 180 sever, 270 východ.

PRŮVODCE HODINOU IV-4

V této hodině se studenti naučí pomocí micro:bitu měřit intenzitu magnetického pole.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty
- Magnet

Postup

Zeptejte se studentů, jaká je jednotka intenzity magnetického pole (magnetické indukce).

Je to Tesla. V praxi se používají její dílčí jednotky. Micro:bit umí pomocí kompasu měřit intenzitu magnetického pole v jednotkách nano Tesla (nT)

Můžeme tedy napsat následující program, který změří hodnotu magnetického pole, a pak zjišťuje, zda absolutní hodnota změny magnetického pole v okolí překročí určitou hodnotu (zde 5000 nT). Pokud ano, tak zobrazí na určitou dobu smajlík.

```
from microbit import *
hodnota = 5000
compass.calibrate()
pocatek = compass.get_field_strength()
while True:
    sleep(100)
    sila = compass.get_field_strength()
    if abs(sila - pocatek) > hodnota:
        display.show(Image.HAPPY)
        sleep(3000)
        display.clear()
```

Vyzkoušejte v okolí, kterých přístrojů se nachází magnetické pole. Např. počítače, mobily, tablety. Rovněž také zmagnetizované nůžky, nože anebo šroubováky.

S pomocí tohoto programu můžete předvést následující kouzlo. V ruce ukryjete malý silný magnet a přejedete touto rukou nad micro:bitem. Micro:bit zobrazí úsměv. Řekněte neznalému, že micro:bit se rozsvítí pouze v okolí lidí s magnetickým potenciálem a nechte je pohyb zopakovat. Bez magnetu samozřejmě k ničemu nedojde.

5 SÍŤ

Tato kapitola se věnuje propojení dvou nebo více Micro:bitů pomocí sítě ať drátové či bezdrátové

Co se naučíte

- Základní principy sítí
- Propojit dva Micro:bity pomocí drátu a přenést informaci
- Totéž bezdrátově, pomocí rádia

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích

Časová náročnost

Jedna až čtyři hodiny po 45 minutách v závislosti na tom, zda a jak podrobně chcete probírat drátové propojení.

Pokud nechcete pracovat s kabely, můžete přeskočit rovnou na hodinu V-4 (bezdrátový přenos).

PRŮVODCE HODINOU V-1

Studenti se v této hodině seznámí s některými pojmy z počítačových sítí. Propojí si micro:bity pomocí kabelu a vyzkouší si přenos dat.

Co bude v této hodině potřeba:

- PC s editorem Mu.
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Programy v celé kapitole jsou vesměs delší. Zvažte proto, zda studenty necháte kód opisovat nebo jim jej vhodným způsobem poskytnete.

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

Vysvětlete studentům pojmy počítačová síť, drátová, bezdrátová. Diskutujte o příkladech. Vysvětlete pojmy *duplex*, *half duplex*, *simplex* a *síťový protokol*.

Nechte studenty, ať se domluví, kdo z nich bude Vysílač (bude vysílat signál) a kdo Přijímač (bude přijímat signál).

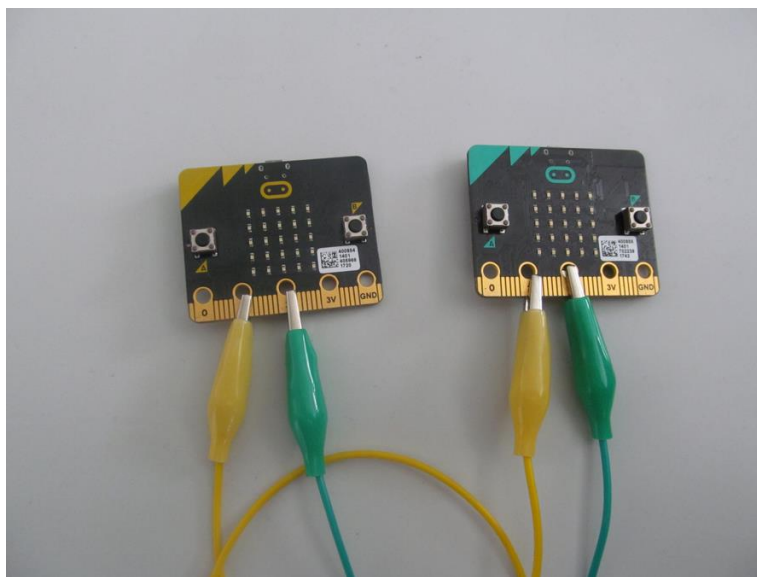
Na micro:bitu označeném *Vysílač* odlaďte následující program:

```
from microbit import *
while True:
    if button_a.was_pressed():
        display.show("A")
        pin1.write_digital(1)
        sleep(1000)
    else:
        pin1.write_digital(0)
    if button_b.was_pressed():
        display.show("B")
        pin2.write_digital(1)
        sleep(1000)
    else:
        pin2.write_digital(0)
    display.clear()
```

Obdobně na micro:bitu *Přijímač*:

```
from microbit import *
while True:
    if pin1.read_digital():
        display.show("A")
    elif pin2.read_digital():
        display.show("B")
    sleep(1000)
    display.clear()
```

Propojte po dvou micro:bity, tak že spojíte (nejlépe vodiči s krokodýlky) vzájemně piny 1 na obou stranách a stejně tak piny 2. Jeden z nich musí být vysílač a druhý přijímač.



Vyzkoušejte přenos signálu.

Vysvětlíte studentům, že skutečnost, jaký signál *Vysílače* bude mít u *Přijímače*, jaký význam, záleží pouze na předchozí domluvě. Jedná se o tzv. *Síťový protokol*.

2. krok 20 minut

Obráťte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu. Po vyzkoušení s nimi prodiskutujte, že se jedná vlastně o paralelní přenos, neboť můžeme přenášet signál po více vodičích současně. Nechte studenty přijít na to, kolik různých možností přeneseného kódu umožňuje daná kombinace – dvě na druhou – čtyři možnosti.

Další možné úlohy:

- Přidejte k danému programu možnost zobrazení C při stisku obou kláves současně.
 - Zkuste nechat naprogramovat přenos čísel 0 až 3 (1 až 4).
- Řešení: Je možné přenést čtyři binární stavy (00, 01, 10, 11). Každému stavu přiřadíte symbol.

PRŮVODCE HODINOU V-2

Studenti si v této hodině vyzkouší přenos signálu mezi dvěma micro:bity pomocí jednoho vodiče – sériový přenos.

Co bude v této hodině potřeba:

PC s editorem Mu.

Micro:bit s USB kabelem

Vodič nejlépe s krokodýlky na obou koncích

Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)

Prezentaci k této lekci

Pracovní listy pro studenty

1. krok 25 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

Vysvětlíte studentům pojem sériový přenos. Řekněte jim, že v této hodině se programy budou týkat sériového přenosu.

Nechte studenty, ať se domluví, kdo z nich bude *Vysílač* (bude vysílat signál) a kdo *Přijímač* (bude přijímat signál).

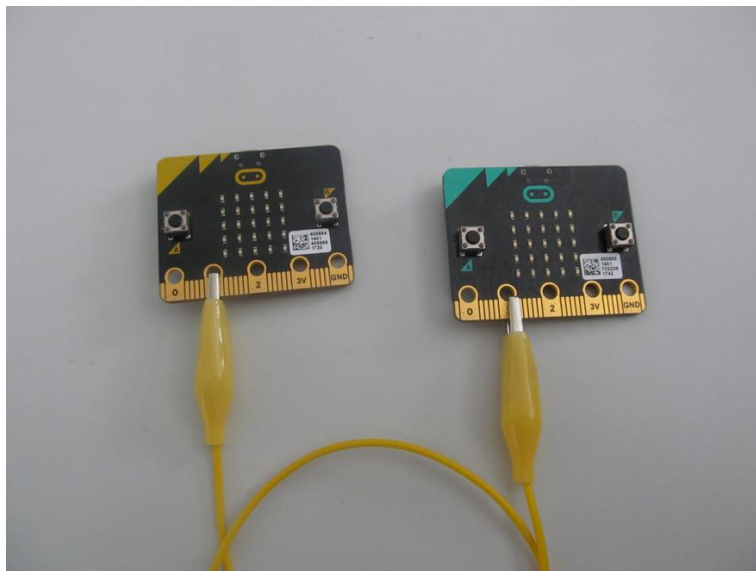
Na micro:bitu označeném *Vysílač* odlaďte následující program:

```
from microbit import *
while True:
    if button_a.was_pressed():
        display.show("A")
        pin1.write_digital(1)
        sleep(500)
        pin1.write_digital(0)
    if button_b.was_pressed():
        display.show("B")
        pin1.write_digital(1)
        sleep(2000)
        pin1.write_digital(0)
    display.clear()
```

Obdobně na micro:bitu *Přijímač*:

```
from microbit import *
while True:
    if pin1.read_digital():
        start = running_time()
        while pin1.read_digital():
            pass
        konec = running_time()
        cas = konec - start
        if cas < 1000:
            display.show("A")
        else:
            display.show("B")
        sleep(1000)
        display.clear()
```

Propojte po dvou micro:bity, tak že spojíte (nejlépe vodiči s krokodýlky) vzájemně piny 1 na obou stranách. Zresetujte micro:bity a vyzkoušejte přenos signálu.



Pokud micro:bit Přijímač zaznamená na pinu1 signál, zjišťuje si jeho délku. Pokud je délka kratší než 1 sekunda, považuje to za první typ signálu (např. binární 0). pokud je delší, pak za druhý typ signálu (např. binární jedna).

2. krok 20 minut

Obrátte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu. Po vyzkoušení s nimi prodiskutujte, jaký význam může mít přenesený signál.

Prodiskutujte možnosti použití daného typu přenosu:

- Morseova abeceda
- ASCII kódy

Jednodušší je samozřejmě přenos ASCII kódu – každý symbol je stejně dlouhý (7 nebo 8 znaků dle typu kódování).

Program s Morseovou abecedou je vyřešen na stránkách s dokumentací MicroPythonu pro micro:bit.

(<https://microbit-micropython.readthedocs.io/en/latest/tutorials/network.html#the-end-result>)

PRŮVODCE HODINOU V-3

Studenti si v této hodině vyzkouší oboustranný přenos signálu mezi dvěma Micro:bity – duplexní přenos.

Co bude v této hodině potřeba:

PC s editorem Mu.

Micro:bit s USB kabelem

Dva vodiče nejlépe s krokodýlky na obou koncích

Pokud je k dispozici, tak dataprojektor

Prezentaci k této lekci

Pracovní listy pro studenty

1. krok 30 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

Vysvětlíte studentům pojem duplexní (obousměrný) přenos. Řekněte jim, že v této hodině budeme informaci přenášet oběma směry.

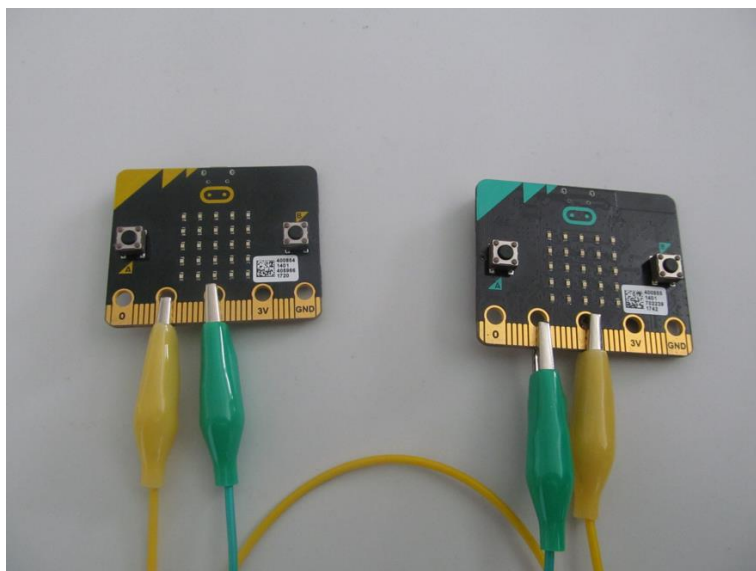
V této hodině budeme na obou micro:bitech používat stejný program. Nechte studenty, ať si naprogramují a odladí následující program:

```
from microbit import *
while True:
    if pin1.read_digital():
        display.show(Image.HAPPY)
    else:
        display.clear()
    if button_a.was_pressed():
        pin2.write_digital(1)
    else:
        pin2.write_digital(0)
    sleep(100)
```

Při úvodním ladění si mohou studenti vystačit i sami. Programy jsou připravené tak, aby vysílali na pinu2 a naslouchali na pinu1.

Pokud si studenti propojí vodičem pin2 s pinem1, bude jim program fungovat i v rámci jednoho micro:bitu a mohou si tak vše vyzkoušet sami na sobě. Zkuste je to na závěr nechat vyzkoušet.

Nyní propojte dva micro:bity tak, že pin2 na jednom připojíte k pinu1 na druhém a naopak. Oba micro:bity připojte ke zdroji energie a zresetujte. Vyzkoušejte přenos signálu.



2. krok 15 minut

Nechte studenty upravit druhy signálu atd. Můžete upravit program tak, že bude rozlišovat druh signálu dle jeho délky. Viz minulá hodina.

Pohovořte o výhodách a nevýhodách duplexního přenosu.

PRŮVODCE HODINOU V-4

Studenti si v této hodině vyzkouší radiový přenos signálu mezi dvěma micro:bity.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Rozdejte studentům micro:bity. Řekněte jim, ať se rozdělí do dvojic.

Vysvětlete studentům pojem radiový přenos. Je možné naladit celkem 84 kanálů označených 0 až 83. Jedná se o frekvenci 2400 MHz (odpovídá kanálu 0), každý kanál má rozsah cca. 1 MHz. Defaultní kanál je 7.

Nechte studenty ať se domluví, kdo z nich bude *Vysílač* (bude vysílat signál) a kdo *Přijímač* (bude přijímat signál). Dále je nechte ať si spolu domluví i kanál. Dbejte na to ať si domluví každá dvojice rozdílný kanál, aby se vzájemně nerušili.

Na Micro:bitu označeném *Vysílač* odlaďte následující program:

```
from microbit import *
import radio
kanal = 23
radio.on()
radio.config(channel = kanal)
while True:
    if button_a.is_pressed():
        radio.send("Zprava")
        sleep(1000)
```

Obdobně na Micro:bitu *Přijímač*:

```
from microbit import *
import radio
kanal = 23
radio.on()
radio.config(channel = kanal)
while True:
    zprava = radio.receive()
    if (zprava):
        display.scroll(zprava)
        zprava = ""
```

Můžete studenty nechat nejprve vyzkoušet programy bez nastavení kanálu anebo je nechte nastavit všechny stejný kanál. Můžete při tom pohovořit o bezpečnosti a typu útoku *man in the middle*.

2. krok 20 minut

Obráťte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu.
Je-li to možné, pusťte studenty z třídy a nechte je vyzkoušet dosah signálu.

6 PŘIPOJOVÁNÍ PERIFÉRIÍ

Tato kapitola se věnuje připojení periférií. Pod pojmem periférie zde rozumíme další součástky, ze kterých budeme získávat data nebo takové, jejichž chování budeme ovládat pomocí micro:bitu. Konkrétně budeme ovládat tříbarevnou diodu a získávat data z externího teplotního čidla.

Kapitola je koncipována jako volitelná, autoři původně chtěli skončit minulou kapitolou, neboť se předpokládalo, že studenti nebudou pro studium této učebnice potřebovat nic jiného než micro:bit. Jelikož jsme však již připojovali sluchátka a propojovali micro:bity vzájemně, je nám líto nezařadit ještě tuto kapitolu. Ponecháváme zcela na vyučujícím, zda se jí rozhodne zařadit. Znamená rovněž další finanční náklady, které však jsou oproti ceně micro:bitu minimální.

Následující úlohy lze samozřejmě řešit i alternativně pomocí nepájivého pole a přiblížit se tak více elektrotechnické praxi. To ponecháváme na rozhodnutí vyučujícího. Pro vyučující, kteří se rozhodnou nepájivé pole používat, jistě nebude problém si úlohy upravit.

Co se naučíte

- Připojit a ovládat tříbarevnou diodu
- Připojit teplotní čidlo a zjistit pomocí něj teplotu
- Pomocí dvou micro:bitů sestavit jednoduchý bezdrátový teploměr s jedním čidlem

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Čtyři vodiče nejlépe s krokodýlky na obou koncích
- Tříbarevnou diodu se společnou katodou (záporným pólem)
- Teplotní čidlo pro napětí 3 V např. TMP-36

Časová náročnost

Dvě až tři vyučovací hodiny po 45 minutách, dle toho, jaké periferie budete připojovat.

PRŮVODCE HODINOU VI-1

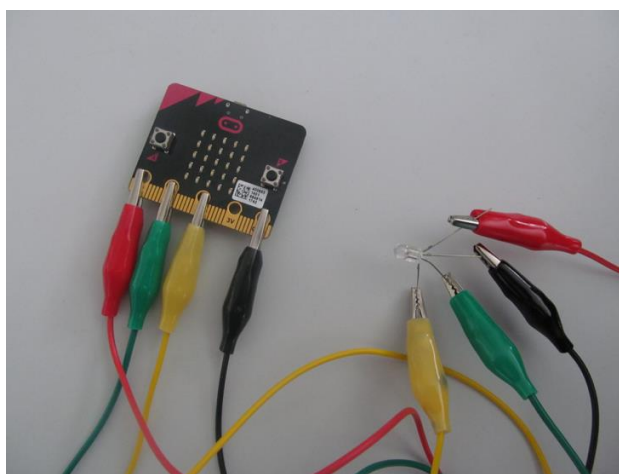
Studenti se v této hodině seznámí s možností připojení externí součástky k micro:bitu, v tomto případě s tříbarevnou diodou. Vyzkouší si digitální i analogový zápis.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Čtyři vodiče nejlépe s krokodýlky na obou koncích
- Tříbarevnou diodu se společnou katodou
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Zapojte tříbarevnou diodu k micro:bitu následujícím způsobem. Zem (GND) zapojte na nejdelší pin tříbarevné diody. Ostatní zapojení je doporučené. pin0 zapojte k samostatnému pinu diody (červená), který je na jedné straně diody. Oba piny z druhé strany diody zapojte tak, že pin blíže ke středu (zelená) připojíte k pinu1 a poslední pin (modrá) k pinu2. Viz obrázek.



Nyní odlad'te a nahrajte následující program:

```
from microbit import *
pin0.write_digital(1)
sleep(2000)
pin0.write_digital(0)
pin1.write_digital(1)
sleep(2000)
pin1.write_digital(0)
pin2.write_digital(1)
sleep(2000)
pin2.write_digital(0)
```

Pokud je vše v pořádku, měly by se postupně rozsvítit vždy na dvě vteřiny postupně červená, zelená a modrá. Jedná se o digitální dvoustavový (binární) zápis, – diody buď zcela svítí anebo nesvítí – zapisujeme jedničku anebo nulu.

Program lze zjednodušit a zpřehlednit:

```
from microbit import *
A = [pin0, pin1, pin2]
for I in A:
    I.write_digital(1)
    sleep(2000)
    I.write_digital(0)
```

Všimněte si konstrukce s polem pinů (proměnná A). Vysvětlete studentům pojem pole, jeho použití a zápis v Pythonu. (Kratší a přehlednější zápis.)

2. krok 20 minut

Nyní si ukážeme při stejném zapojení postupné rozsvěcení a zhasínání jedné barvy (té která je připojená na pin0), které použijeme v následujícím příkladě. V tomto případě se jedná o *diskreditaci analogového zapojení*, k dispozici máme 1024 stavů. Vysvětlete to studentům.

```
from microbit import *
while True:
    for I in range(0, 1024):
        pin0.write_analog(I)
        sleep(2)
    sleep(1000)
    for I in range(1023, -1, -1):
        pin0.write_analog(I)
        sleep(2)
```

V příští hodině budete vyrábět magickou lampu. Je vhodné, aby se studenti na tuto výrobu připravili. Je třeba připravit vhodné stínítko ať třeba jako papírový válec anebo vyrobené na 3D tiskárně. Je rovněž možné použít lampion.

PRŮVODCE HODINOU VI-2

Studenti v této hodině sestojí pomocí tříbarevné diody tzv. magickou lampu. To jest lampu, která postupně mění tři barvy, přičemž vytváří různé odstíny. Téma je na celou hodinu, proto jej dále nedělíme.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Čtyři vodiče nejlépe s krokodýlky na obou koncích
- Tříbarevnou diodu se společnou katodou
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok – celá hodina

Ponechte zapojení jako v minulé hodině a odlaďte a nahrajte následující program:

```
from microbit import *
import random
A = [pin0, pin1, pin2]
minula = 2
while True:
    barva = random.randint(0, 2)
    while (barva == minula):
        barva = random.randint(0, 2)
    delka = random.randint(1000, 5000)
    for I in range(0, 1024):
        A[barva].write_analog(I)
        A[minula].write_analog(1023-I)
        sleep(2)
    sleep(delka)
    minula = barva
```

Jedná se o program zvaný „Magická lampa“. Náhodně postupně rozsvěcí jednu z tří možných barev. Pak jí postupně zhasíná a současně rozsvěcí jinou. Pro zjednodušení je opět použita konstrukce s polem pinů. Proměnná minula hlídá, jaká barva byla rozsvícená minule, aby došlo ke změně barvy. Upozorněte studenty, že jak barva tak délka svitu jsou voleny pomocí generátoru náhodných čísel.

Nyní nechte studenty vyrobit skutečnou lampu. Například jenom jako váleček ze čtvrtky, kde jednotlivé piny prostrčíte čtvrtkou ven. Nebo vytiskněte nějakou lampu pomocí 3D tiskárny. Nebo použijte nějaký lampion.

Důležité je, aby se jednotlivé konektory nedotýkaly.

Lze také použít dva micro:bity, jeden ponechat na ovládání lampy a druhým mu vzdáleně posílat pokyny. Délka svitu, druhy barev, zapnout vypnout atd.

PRŮVODCE HODINOU VI-3

Studenti se v této hodině seznámí s možností připojení externí součástky k micro:bitu. V tomto případě se setkají s teplotním čidlem. Vyzkouší si analogový vstup.

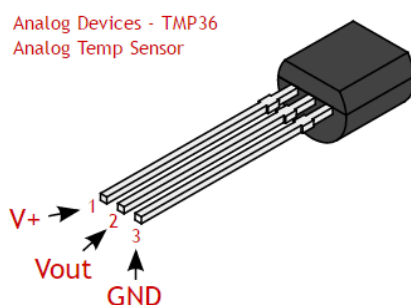
Co bude v této hodině potřeba:

- PC s editorem Mu.
- Micro:bit s USB kabelem
- Tři vodiče nejlépe s krokodýlky na obou koncích
- Levné teplotní čidlo pracující s napětím 3 V (např. TMP36). K němu potřebujete schéma zapojení (datasheet)
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

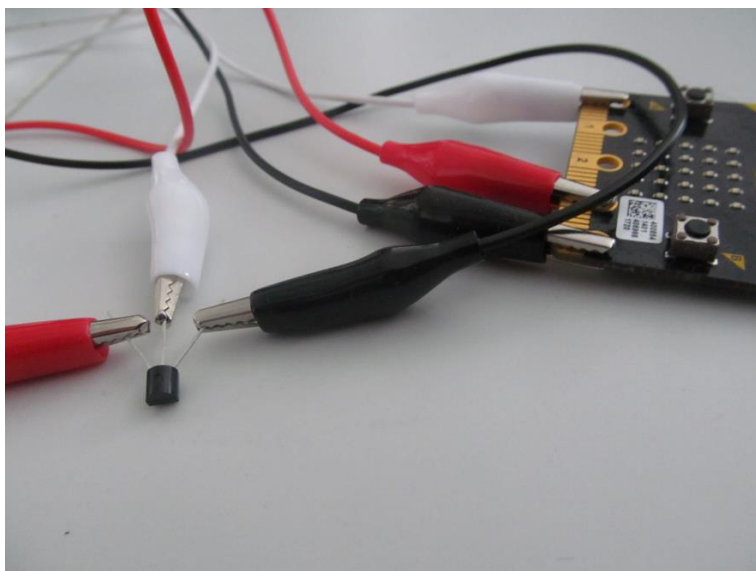
1. krok 10 minut

Pro toto cvičení potřebujete nějaké levné teplotní čidlo pracující s napětím 3 V. Zde je použito čidlo 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 nebo odjinud z internetu.

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



Zde V+ je napájení, připojte na něj 3V, GND (zem) připojte na GND a Vout je výstup, který zapojte na libovolný pin, například na pin0. Zapojení je ukázáno na následujícím obrázku. Všimněte si na fotografii, že plochá hrana čidla je dole (tedy opačně než na předchozím obrázku). Dejte si pozor abyste nespletli zapojení napájení a země, mohli byste snadno teplotní čidlo zničit. Nechte studenty čidlo zapojit a pak jim pro jistotu zapojení zkontrolujte.



2. krok 10 minut

Vysvětlíte studentům princip funkce čidla.

Čidlo po připojení napájení a země začne měřit teplotu a výsledek sděluje úrovní napětí na výstupním pinu (Vout). Je zde napětí od 0 do 1023 mV. Toto napětí ukazuje procento ze vstupního napětí, které je u micro:bitu 3.18 V, jak můžete ověřit Voltmetrem.

Proto je výpočet napětí:

$$napeti = \frac{Vout \cdot 3.18}{1024}$$

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 datasheetu vašeho čidla.

3. krok 15 minut

Nechte studenty přepsat a odladit program:

```
from microbit import *
while True:
    hodnota = pin0.read_analog()
    napeti = hodnota * (3180 / 1024)
    teplota = (napeti - 500) / 10
    display.scroll(round(teplota, 1))
    sleep(10000)
```

Pauza mezi jednotlivými měřeními je 10 sekund. Tu si samozřejmě studenti mohou upravit, dle vlastního přání.

Rovněž si všimněte zaokrouhlení na jedno desetinné místo pomocí funkce `round()`.

Vysvětlíte studentům, že po zapojení chvíli trvá, než se teplotní čidlo srovná na teplotu měřeného okolí. Zejména pokud jej před použitím drželi delší dobu v ruce. První dva až tři výsledky doporučujeme ignorovat. Teplota se postupně ustaluje na určité hodnotě.

Nechte studenty ověřit měřenou teplotu vzájemně a pokud můžete i s jiným teploměrem.

Pokud se výsledky významně liší, zkuste některý z následujících kroků:

- ověřte program, zejména výpočty
- 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 na výstupech micro:bitu (může být např. 3,17 V)

4. krok 10 minut

Micro:bit obsahuje funkci pro měření teploty procesoru. Je možné ji použít i k měření teploty, ale počítejte, že zejména při delším měření může ukazovat vyšší teplotu, než je ve skutečnosti, díky ohřívání čidla procesorem. Program, který jej využívá, by pak vypadal asi takto:

```
from microbit import *
while True:
    teplota = temperature()
    display.scroll(teplota)
    sleep(10000)
```

Zkuste porovnat naměřené teploty oběma způsoby. Pokud máte více micro:bitů, můžete mít spuštěny současně oba programy a porovnávat teploty. Anebo zobrazujte jednotlivé teploty střídavě či po stisku tlačítek.