ROBOTIKA PRO STŘEDNÍ ŠKOLY: PROGRAMUJEME MICRO:BIT POMOCÍ PYTHONU

PRACOVNÍ LISTY

V tomto souboru naleznete všechny pracovní listy vztahující se k této učebnici. Pokud chcete poskytnout studentům pracovní listy ve formátu PDF, pak můžete buď v prohlížeči PDF (pravděpodobně Acrobat Reader) uložit potřebné stránky pomocí možnosti tisku do PDF anebo navštívit webové stránky:

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

kde si můžete jednotlivé pracovní listy stáhnout. 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 tohoto 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 „ucebnice“. 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.

PRACOVNÍ LIST I-1

První seznámení s *micro:bitem* a programátorským editorem *Mu*. Vytvoření prvního programu, který na displej micro:bitu napíše text „Ahoj svete“.

Co se naučíte

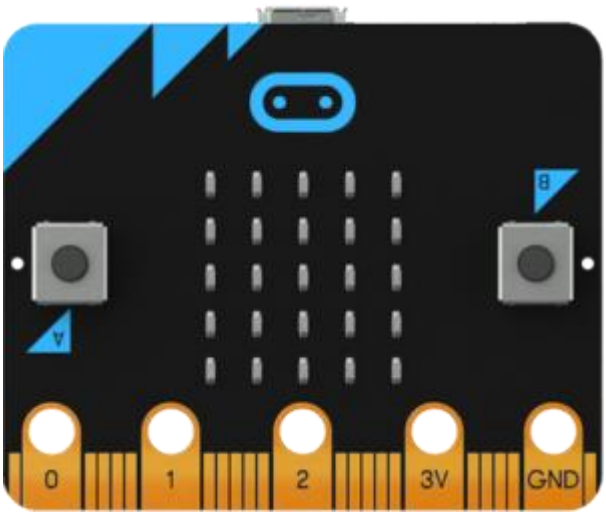
- Základu práce s micro:bitem
- Práci s editorem Mu
- Odladění prvního programu

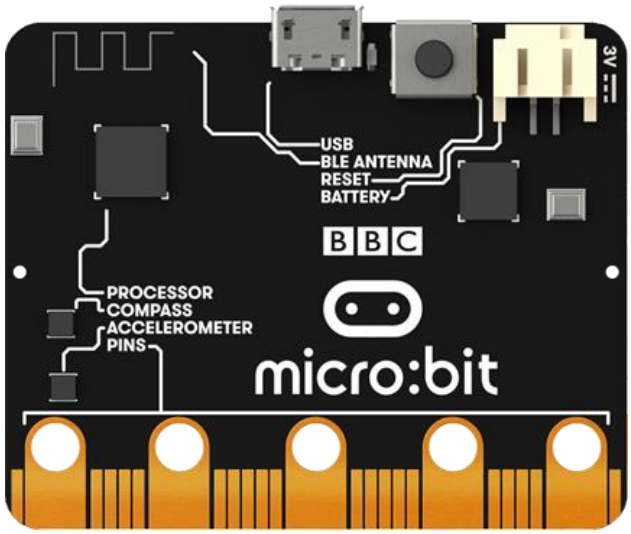
Co budete potřebovat

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

A jděte na to …

Prohlédněte si dobře micro:bit.





Na přední straně (na obrázku vlevo) se nachází matice 5x5 LED diod a dvě programovatelná tlačítka označená A a B. Ve spodní části se nachází 17 **GPIO** (general-purpose input/output) pinů, z nich tři jsou s velkými konektory označené 0, 1 a 2 stejně tak jsou s velkými konektory výstup 3 V a zem označená GND.

Micro:bit má dva vstupy a to micro USB a napájecí vstup pro baterry pack. Napájený může být i z PC přes USB kabel.

Na zadní straně je mezi těmito vstupy tlačítko RESET. Po jeho stisku se micro:bit chová, jako bychom jej znovu spustili. Na zadní straně si rovněž všimněte popisu součástek.

Nyní spusťe program *Mu*. Význam tlačítek v menu nahoře je:

- New, Load, Save – Nový program, načtení uloženého programu, uložení (při prvním se ptá na jméno)
- Flash – Přeložení programu do zdrojového kódu a jeho nahrání na micro:bit
- Files – zobrazí soubory na micro:bitu a soubory (programy) v domovské složce programu
- Repl – možnost zkoušet příkazy přímo na micro:bitu bez psaní programu, 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.
- Help – odkaz na stránky s nápovědou
- Quit – konec

Zapište do programu dva řádky kódu :

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

Připojte micro:bit a vyčkejte, než se zobrazí micro:bit jako připojené zařízení. Program uložte a pojmenujte a pak stiskněte tlačítko Flash. Pokud je vše v pořádku program se nahraje do micro:bitu (po dobu nahrávání bliká žlutá dioda na zadní straně). Pokud se místo automatického nahrání zobrazí okno pro uložení souboru, vyberte micro:bit jako periferní zařízení a uložte jej tam. Nyní by měl po displeji micro:bitu proběhnou text Ahoj svete.

Možné chyby

Program nelze nahrát – zkontrolujte, zda je micro:bit připojený, zkuste pro nápravu v tomto pořadí jiný kabel, jiný USB port , jiný micro:bit, jiný počítač.
Microbit píše něco jiného – jedná se o informaci o chybě. Můžete rovněž zkusit chybu nalézt stiskem klávesy Check v editoru Mu.

Další program

Máte-li hotovo můžete vyzkoušet ještě tento program:

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

Tento program v nekonečné smyčce vypisuje text „Ahoj svete“, vyčká 1 vteřinu a opakuje. Zápis nekonečné smyčky je na řádku 2. Na řádku 4 je příkaz, že program má čekat 1000 tisícin vteřiny – tedy jednu vteřinu.

Pozor na chyby:

- Příkazy na řádku 3 a 4 musí být odsazeny zleva přesně o čtyři mezery. Mu vám to bude už nabízet. Nelze použít jiný počet mezer nebo tabelátor, jak bývá někdy možné v jiných verzích Pythonu.
- Na konci řádku 2 je dvojtečka, na to začátečníci často zapomínají

Důležité webové adresy

Kde najít rady nebo materiál pro další studium:

Domácí stránka micro:bitu

https://microbit.org/

Dokumentace k MicroPythonu:

https://microbit-micropython.readthedocs.io/en/latest/

Editor Mu:

https://codewith.mu/

PRACOVNÍ LIST I-2

V této hodině se naučíte používat **cykly** a ukážete si další způsoby výpisu informací na displej micro:bitu.

Co se naučíte

- Nekonečnou smyčku
- Cykly *for* a *while*
- Výpis znaku a smazání obrazovky

Co budete potřebovat

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

A jděte na to …

Zapište následující program do editoru *Mu* a nahrajte jej do micro:bitu.

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

Jedná se o nekonečnou smyčku – výpis se opakuje.

Pozor na syntaxi:

- True musí být s velkým T
- Na konci zápisu cyklu je dvojtečka
- Odsazení musí být o přesně čtyři znaky – jste-li zvyklí z jiných verzí Pythonu na tabelátor, zde jej není možné použít

Nyní řešte úlohu – výpis čísel od jedné do desíti na displej. Použijte postupně dva různé postupy – pomocí cyklu `for` a pomocí cyklu `while`.

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

Zde je použit cyklus `for`. Zápis: `i in range(1, 11)` znamená – za i dosazuj čísla od jedné do desíti.

Pozor jedná se o interval <1,11) nalevo uzavřený a napravo otevřený.

Pozor za čárkou v intervalu musí být mezera.

Otázky:

Jaký je rozdíl mezi řetězcem (stringem) a celým číslem (integerem)?

Nyní tentýž program zapsaný pomocí cyklu `while`:

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

Co znamená negace?

Je totéž `(i > 11)` a not `(i < 11)`?

Který ze zápisů, s `while` nebo s `for`, je vám bližší? Proč?

Na závěr ještě vyzkoušejte následující program, na kterém si vyzkoušíte další funkce:

```
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 příkazu `display.clear()`. Příkaz rovněž umí zobrazit číslo či řetězec. Rozdíl oproti příkazu `display.scroll()` je ten, že poslední znak zůstane zobrazen.

PRACOVNÍ LIST I-3

V této hodině se seznámíte s možností zobrazení jednoduchých obrázků na displeji micro:bitu. Nejprve si ukážete zobrazení připravených obrázků. Pak si zkusíte sestavit a zobrazit obrázek vlastní.

Co se naučíte

- Zobrazení připravených obrázků
- Sestrojení vlastního obrázku

Co budete potřebovat

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

A jděte na to …

Zapište a odlaďte následující kód:

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

Konstrukce `Image.SAD` atd. jsou připravené konstanty – obrázky. Poproste vyučujícího, ať vám poskytne seznam obrázků nebo jej hledejte na webové stránce dokumentace k *MicroPythonu*.

Zkuste ještě následující příklad simulující 100 úderů srdce:

```
from microbit import *
for i in range(1, 100):
    display.show(Image.HEART)
    sleep(400)
    display.show(Image.HEART_SMALL)
    sleep(400)
display.clear()
```

Otázky:

Přemýšlejte, proč je použit příkaz sleep(400)?

Nyní zkuste následující příklad, který vytvoří na displeji obrázek rakety:

```
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 apostrofy a uvnitř končí dvojtečkou
- Čísla od 0 do 9 znamenají intenzitu světla (0 – nesvítí, 9 – svítí naplno)

Je možná použít též následující syntaxi:

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

Vyzkoušejte si sestavit vlastní obrázek.

Důležitá webová adresa

Generátor obrázků:

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

Nutno nastavit matici 5x5 a jazyk Python

PRACOVNÍ LIST I-4

- V této hodině se seznámíte s možností vytvoření jednoduché animace na displeji micro:bitu a dále se naučíte rozsvěcet konkrétní diodu s požadovanou intenzitou světla.

Co se naučíte

- Vytvořit animaci
- Poznáte datovou strukturu list (seznam)
- Rozsvítit konkrétní diodu s požadovanou intenzitou
- Práci s generátorem náhodných čísel
- Zjistit intenzity svícení konkrétní diody

Co budete potřebovat

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

A jděte na to …

Zapište a odlaďte následující kód (anebo jej stáhněte a otevřete dle pokynů vyučujícího):

```
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)
```

Jedná se o jednoduchou animaci startující rakety, vycházející z minulé lekce. Je to vlastně šest obrázků, které se zobrazí příkazem `display.show(raketa, delay=500)` po půl sekundě.

Datová struktura `raketa` je **list** (**seznam**) – jedná se o uspořádanou n-tici, u které záleží na pořadí a umožňuje opakovaný výskyt jednotlivých prvků.

Je možné vypustit obrázek raketa6? Pokud ano jak upravíte program?
Zkuste si vytvořit vlastní animaci.
Nyní zkuste napsat a odladit následující program, který náhodně rozsvěcí diody s různou intenzitou a simuluje tak hvězdnou oblohu:

```
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)
```

V programu je použit generátor náhodných čísel. Ten se nastaví zavedením knihovny import random. Příkaz random.int(A, B) pak vrátí náhodné celé číslo z uzavřeného intervalu A,B.

Příkaz display.set_pixel(X, Y, intenzita) nastaví diodu na souřadnici X,Y na hodnotu intenzita. Intenzita je celé číslo z uzavřeného intervalu 0,9 s významem od 0 – nesvítí do 9 – svítí naplno. Souřadnice X je sloupec (0 až 4 zleva) a Y řádek (0 až 4 shora). Levý horní bod je 0,0 a pravý dolní 4,4.

- Jak pracuje generátor náhodných čísel?
- Jedná se o digitální či analogové zobrazení? (Pracuje úloha s dvěma či více hodnotami?)

Nyní si ukážeme jiný příklad:

```
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)
```

Zde se jedná o čistě digitální zobrazení. Každá dioda nabývá dvou hodnot svítí (intenzita 9) nebo nesvítí (intenzita 0). Funkce `display.get_pixel(x, y)` zjišťuje, zda dioda na souřadnicích X,Y svítí či nikoliv. Pokud vrátí hodnotu 1 (podmínka splněna) dioda se rozsvítí jinak zhasne.

Pozor na dvojí úroveň odsazení. Ve druhé úrovni (u `if – else`) to musí být 8 znaků (násobek 4).

PRACOVNÍ LIST II-1

Ukázka programového větvení pomocí stisku programovatelných tlačítek A a B.

Co se naučíte

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

Co budete potřebovat

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

A jděte na to …

Prohlédněte si dobře micro:bit. Zaměřte svou pozornost na tlačítka. Kolik jich najdete a jaký je jejich význam?

Nyní zapište, odlaďte a nahrajte do micro:bitu následující příklad:

```
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()
```

Pozor na správná odsazení. Odsazení na druhé úrovni (pod `if`) musí být o čtyři mezery oproti první úrovni, celkem tedy 8 mezer.

Které příkazy a jak testují stisk tlačítek?

Existuje i příkaz `button_a.was_pressed()` - ten vrací informaci, zda tlačítko bylo stisknuté od minulé kontroly nebo od začátku programu, pokud jeho stisknutí nebylo dosud kontrolováno.

Nyní si vyzkoušíte práci s oběma tlačítky současně. Odlaďte následující program:

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

Co tento program dělá?

Jaký je význam logické spojky `and`?

V předchozím programu nahraďte řádek spojku `and` spojkou `or` a nahrajte program do micro:bitu:

```
if (button_a.is_pressed()) or (button_b.is_pressed()):
```

Jaká je změna ve funkci programu?

Jaký je význam logické spojky `or`?

Abyste pochopili funkci `get_presses()` zapište a odlaďte následující program:

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

Program po spuštění čeká 10 vteřin. Během této doby opakovaně stiskněte klávesu A. Program poté zobrazí počet vašich stisků.

Neřešený závěrečný příklad: Naprogramujte postřehovou hru. Na micro:bitu se bude střídavě zobrazovat náhodně 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 stisků. Na závěr může být vyhodnocení např. procentem úspěšných pokusů. Pro volbu A nebo B použijte generátor náhodných možností, který znáte z kapitoly 1.

PRACOVNÍ LIST III-1

Co se naučíte

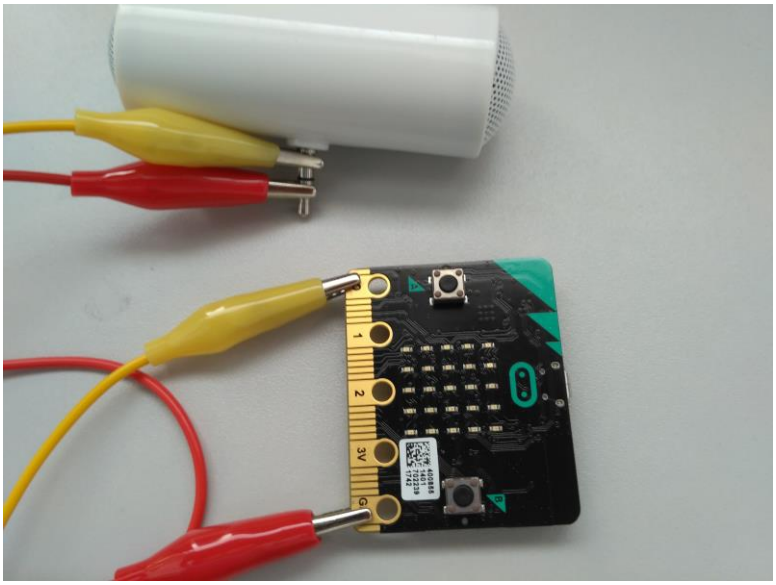
- Připojit k micro:bitu hardware na výstup zvuku
- Přehrát předpřipravenou melodii a zkombinovat jí se zobrazením obrázku

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
- Reproduktor nebo sluchátka s jackem, popřípadě piezzo buzzer.

A jděte na to …

Připojte k micro:bitu sluchátka nebo repráček dle následujícího obrázku:



Připojení sluchátek či reproduktoru je velmi snadné. Budete nyní potřebovat dva vodiče nejlépe opatřené na koncích krokodýly. Ty na dolní straně micro:bitu připněte jeden na GND a druhý na 0. Druhý konec vodičů připojte na jack libovolného reproduktoru či sluchátek. Nezáleží na pořadí, který vodič kam připojíte. Doporučení je následující: má-li váš jack tři vstupy, pak jeden z vodičů připojte na prostřední a druhý na libovolný z krajních vodičů, má-li čtyři vstupy, pak by měly fungovat buď oba krajní nebo oba vnitřní (možná budete muset trochu experimentovat).

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

Nyní nahrajte do micro:bitu následující program:

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

Příkaz na řádku 2 zavádí knihovnu pro práci se zvukem a na řádku 3 se přehraje připravený zvuk. Tento zvuk je celkem dlouhý, a tak je vhodný pro testování.

Seznam všech připravených melodií vám poskytne vyučující.

Pokud máte program v pořádku nahrán na micro:bitu nasaďte si sluchátka. Pokud neslyšíte tón stiskněte tlačítko reset na micro:bitu. Pokud ani nyní nic neslyšíte zkuste jiné konektory na jacku sluchátek. Můžete zkusit místo výstupního pinu 0 na micro:bitu piny 1 nebo 2. Pokud to nepomůže, zkuste jiná sluchátka či jiný micro:bit.

Nyní si zkombinujeme vše, co už znáte z předchozích hodin. Zobrazení obrázku, stisk tlačítek a přehrání zvuku. Nahrajte následující kód do micro:bitu a vyzkoušejte:

```
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()
```

Jaký je význam jednotlivých řádků?

Zkuste si program upravit použitím jiných obrázků a melodií.

PRACOVNÍ LIST III-2

Naučíte se na micro:bitu přehrát vlastní melodii a naučíte jej mluvit.

Co se naučíte

- Naučíte micro:bit mluvit
- Vytvořit vlastní melodii pomocí not a přehrát jí.

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
- Reproduktor nebo sluchátka s jackem, popřípadě piezzo buzzer

A jděte na to …

Napište a odlaďte následující program:

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

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

Na řádku 2 se zavádí knihovna pro hovor a na řádku 4 je zadán příkaz pro mluvení. Zde micro:bit pozdraví. Parametr speed=100 je nepovinný a je možné jej vynechat včetně čárky. (Defaultní hodnota je 72, ale při této hodnotě mluví micro:bit příliš rychle. Čím vyšší číslo je zadáno, tím je řeč pomalejší a naopak.)

Pozor micro:bit mluví pouze anglicky a je tak nutno použít anglickou transkripci. Např. „Josef“ je třeba napsat jako „Yoseph“ atd. A samozřejmě nelze použít české znaky.

Pokud se vám zdá, že micro:bit mluví potichu, zkuste zapojit sluchátka mezi 0 a 1.

Zkuste naučit micro:bit říkat své jméno a příjmení (bez háčků a čárek).

Připojte si opět sluchátka (nebo jiné zvukové zařízení) k micro:bitu mezi 0 a GND a pak přeložte a odlaďte následující program:

```
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)
```

Program by měl hrát melodii „Ovčáci čtveráci“. Pokud máte hudební sluch a vyznáte se v notách, můžete melodii zkusit upravit. Význam jednotlivých tónů je: C4:4 znamená nota C ve čtvrté oktávě (0 – nejnižší, 8 – nejvyšší) o délce 4. Nota R znamená pauzu (rest) o dané délce. Příkaz `music.play(nota)` pak daný záznam přehraje.

Otázka: Co je za strukturu `nota`?

Zkuste si naprogramovat vlastní melodii nebo nějakou známou skladbu.

PRACOVNÍ LIST IV-1

Co se naučíte

- Co je to akcelerometr a jak funguje
- Sledovat natočení micro:bitu v prostoru
- Využít akcelerometr jako ovladač micro:bitu

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
- Reproduktor nebo sluchátka s jackem

A jděte na to …

Pokud neznáte pojem akcelerometr, nechte si jej vysvětlit vyučujícím.

K jakým účelům byste použili akcelerometr?

Napište a odlaďte následující program:

```
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 sleduje náklon micro:bitu dle osy *x* (vlevo a vpravo). Proměnná `mez` určuje, jaký náklon budeme považovat za mezní, abychom řekli, že je micro:bit nakloněn vpravo či vlevo. Experimentujte s tím jak se micro:bit chová dle orientace v prostoru při různém natočení. Zkuste nahradit `get_x` za `get_y` popřípadě `get_z` abyste vyzkoušeli orientaci vůči ose *y* nebo *z*.

Nyní si vyzkoušíte simulaci hudebního nástroje theremin. Pokud nevíte, co je Theremin, zeptejte se vyučujícího nebo si jej najděte na internetu.
Napište a nahrajte 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 sluchátka (repráčky) podobně jako v minulé hodině (mezi piny 0 a GND). Měli byste nyní slyšet tón. Otáčením micro:bitu vlevo a vpravo měníte výšku tónu, od sebe k sobě jeho délku.

Experimentujte se změnou rozsahů (zvětšení či zmenšení tónového rozsahu).

Všimněte si v programu, jakým způsobem se v Pythonu spojují dva řetězce. Jedná se vlastně o sčítání.

PRACOVNÍ LIST IV-2

Co se naučíte

- Co jsou to gesta
- Jaká gesta umí micro:bit zaznamenat
- Jak na micro:bitu ovládat gesta a jak je použít.

Co budete potřebovat

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

A jděte na to …

Otázky:

Co jsou to gesta? Jaké znáte?

Prohlédněte si seznam gest. Jak byste mohli jednotlivá gesta využít?

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, je-li, jakkoliv nakloněn, se zobrazí zamračená tvář.

Vyzkoušejte i jiná gesta a jiné smajlíky. Pozor na zkoušení volného pádu atd. Nerozbijte si micro:bit. Můžete přidat časovou pauzu např. 3 sekundy po zobrazení gesta, abyste jej stihli.

Zkuste program upravit tak, aby zobrazoval úsměv anebo nic.

Nyní napište (nahrajte) a odlaďte následující program. Jedná se o variaci hry Magic 8 ball.

Myslete na nějakou otázku, která vás trápí a pak zatřeste micro:bitem a on vám na vaši otázku odpoví. Jak program funguje?

Zkuste přidat další možnosti. Pozor, jak již víte, nesmíte používat háčky a čárky.

```
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 vnitrni ja rika 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 `accelerometr.was_gesture()`, která bere v úvahu pouze gesta od posledního volání.

PRACOVNÍ LIST IV-3

Co se naučíte

- Jak pracovat s micro:bitem jako s kompasem
- Co je to azimut a jak jej pomocí micro:bitu stanovit

Co budete potřebovat

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

A jděte na to …

Vysvětlete pojmy kompas a azimut.
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 vyzkoušet 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. Tuto **kalibraci musíme provést před každým použitím kompasu**. Micro:bit položte na rovnou plochu nebo jej držte co nejvíce rovně. Micro:bit nyní ukáže na displeji azimut (úhel od severu). Směr azimutu je přímo od displeje nahoru.

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 12 směr nahoru ukazuje hodnota 0. Můžete si to ověřit následujícím prográmkem:

```
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])
```

Otázky a úkoly:

Lze nahradit číslo 15, číslem 375? Jak musíme program upravit?

Můžeme místo % 12 napsat + 12? Jak musíme program upravit?

Upravte program pro zobrazení směru na sever 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.

PRACOVNÍ LIST IV-4

Co se naučíte

- Pomocí micro:bitu měřit intenzitu magnetického pole

Co budete potřebovat

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

A jděte na to …

Otázka: Co je to za měrnou jednotku nT (nano Tesla) a pro co se používá?

Můžeme 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.

PRACOVNÍ LIST V-1

Co se naučíte

- Co je to počítačová síť, jaké jsou typy sítě
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

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

A jděte na to …

Rozdělte se do dvojic.

Popovídejte si s vyučujícím o tom, co jsou počítačové sítě a jaké jsou jejich typy.

Nyní se domluvte, kdo ve dvojici bude *Vysílač* (bude signál vysílat) a kdo *Přijímač* (bude signál přijímat).

Vysílač odladí na micro:bitu 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()
```

Přijímač odladí následující program:

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

Propojte nyní micro:bity "Přijímač" a "Vysílač" dvěma kabely s krokodýlky. Vzájemně propojíte na obou stranách piny1 a piny2. Micro:bity připojte ke zdroji energie a pro jistotu resetujte a vyzkoušejte přenos signálu po stisku tlačítek A nebo B na Vysílači.

Vyměňte si role a zopakujte si zadání v opačných pozicích.

Jedná se o paralelní přenos signálu – vysvětlete si tento pojem.

- Kolik stavů můžeme přenést při tomto zapojení?
- Jak byste upravili programy, abyste přenesli i písmeno C?
- Kolik různých stavů je teoreticky možné takto mezi dvěma micro:bity přenášet?

PRACOVNÍ LIST V-2

Co se naučíte

- Sériový přenos
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

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

A jděte na to …

Rozdělte se do dvojic a domluvte se kdo ve dvojici bude *Vysílač* a kdo *Přijímač*.
Vysílač odladí na micro:bitu 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()
```

Přijímač odladí následující:

```
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 nyní micro:bity kabelem s krokodýlky. Vzájemně propojíte na obou stranách piny1. Micro:bity připojte ke zdroji energie a pro jistotu resetujte a vyzkoušejte přenos signálu.

Vyměňte si role a zopakujte si zadání v opačných pozicích. Jak pozná *Přijímač*, o jaký signál se jedná?

Jedná se o sériový přenos signálu – vysvětlete si tento pojem. Napadá vás druh kódování, které lze tímto způsobem přenášet?

PRACOVNÍ LIST V-3

Co se naučíte

- Obousměrný přenos
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

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

A jděte na to …

Rozdělte se do dvojic. Každý si odlaďte na svém micro:bitu 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)
```

Je-li program odladěn, můžete propojit dva micro:bity, tak že pin1 na jednom propojíte s pinem2 na druhém a naopak. Na pin2 se vysílá na pinu1 naslouchá.

Vyzkoušejte obousměrný přenos.

Jaké jsou výhody a nevýhody tohoto přenosu?

Program můžete vyzkoušet i na jednom micro:bitu, pokud u něj propojíte kabelem pin1 s pin2.

PRACOVNÍ LIST V-4

Co se naučíte

- Sériový přenos
- Propojit dva micro:bity rádiovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

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

A jděte na to …

Rozdělte se do dvojic a domluvte se, kdo ve dvojici bude *Vysílač* a kdo *Přijímač*. Dále je nutné domluvit si kanál, na kterém si budete povídat. Kanál je číslo od 0 do 83. Pokud jej neuvedete použije se defaultní kanál 7. Pokud si domluvíte stejný kanál více dvojic, budete se vzájemně rušit. *Vysílač* odladí na micro:bitu 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)
```

Přijímač odladí následující program:

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

Vyzkoušejte přenos signálu. K čemu slouží nastavení kanálu (channel)?

Vyměňte si role a zopakujte si zadání v opačných pozicích.

Pohovořte si o možné bezpečnosti přenosu.

Odhadněte, co znamená *Man in the middle* attack.

PRACOVNÍ LIST VI-1

Co se naučíte

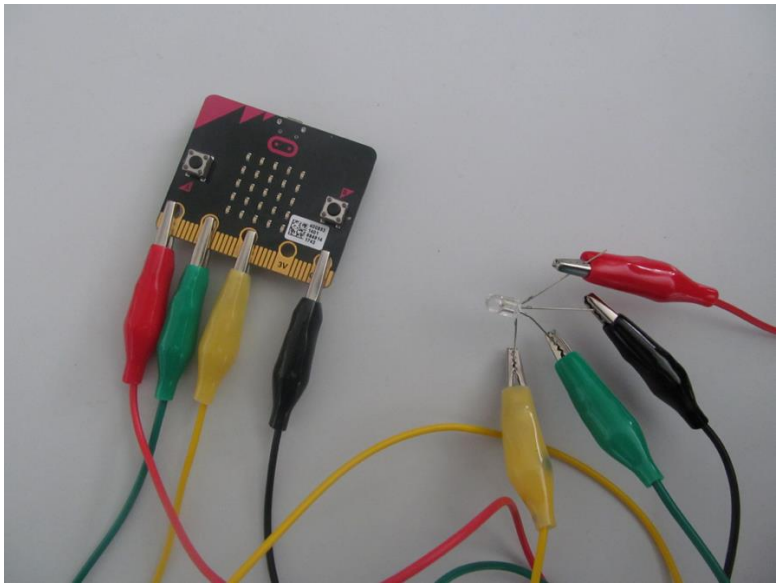
- Co je to tříbarevná dioda a jak jí připojit k micro:bitu
- Digitální i analogový výstup na periférie

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
- Trojbarevnou diodu se společnou katodou

A jděte na to ...

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 odlaďte 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 zápis – diody buď zcela svítí anebo nesvítí. Zapisujeme jedničku anebo nulu. Program nyní zjednodušíme:

```
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. Tuto konstrukci použijeme proto, abychom se mohli obracet na prvek pole pinů a nemuseli vždy vypisovat konkrétní pin.

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.

```
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. Domluvte se s vyučujícím na vhodném materiálu pro její stínítko.

PRACOVNÍ LIST VI-2

Co se naučíte

Jak vzájemně měnit všechny barvy na tříbarevné diodě

Sestrojit magickou lampu

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

Trojbarevnou diodu se společnou katodou

A jděte na to ...

Při stejném zapojení jako v minulé hodině 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. Jak barva tak délka svitu jsou voleny pomocí generátoru náhodných čísel.

Na závěr můžete vyrobit skutečnou lampu. Například jenom jako váleček ze čtvrtky, kde jednotlivé piny prostrčíte čtvrtkou ven. Tím současně dosáhnete toho, že se jednotlivé konektory nebudou dotýkat.

Nyní můžete sestrojit magickou lampu. Lampu můžete vyrobit z papíru (třeba jen jako válec) anebo použít nějakou konstrukci vytištěnou na 3D tiskárně. Můžete také použít nějaký lampion.

PRACOVNÍ LIST VI-3

Co se naučíte

- Jak pomocí micro:bitu a jednoduchého teplotního čidla měřit teplotu
- Zpracovat analogový vstup

Co budete potřebovat

PC s nainstalovaným editorem Mu

Propojovací USB kabel s micro USB koncovkou

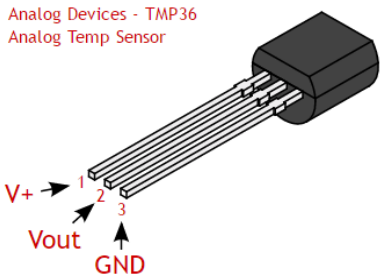
Micro:bit

Tři vodiče nejlépe s krokodýlky na obou koncích

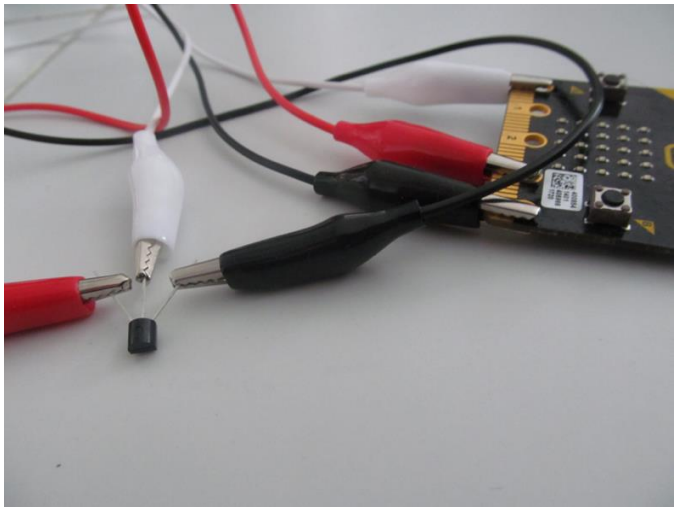
Teplotní čidlo pro napětí 3V, nejlépe TMP 36

A jděte na to …

Zapojte dle následujícího schématu a fotografie:



Zde V+ je napájení, připojte na něj 3 V, GND (zem) připojte na GND a Vout je výstup, který zapojte na libovolný pin, například na pin0:



Všimněte si na fotografii, že rovná strana je dole (čidlo je opačně než na obrázku). Dejte si pozor abyste nespletli (nepřehodili) zapojení napájení a země, mohli byste snadno teplotní čidlo zničit.

Čidlo po připojení napájení a země začne měřit teplotu a výsledek sděluje úrovní napětí na výstupním pinu (Vout), kde může být napětí od 0 do 1023 mV. Toto napětí vlastně ukazuje procento ze vstupního napětí, které je u Micro:bitu 3.18 V.

Proto pro výpočet napětí platí následující vzorec:

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

Odtud pak již vypočteme teplotu (ve stupních celsia):

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

Tento vzorec je dán dokumentací k teplotnímu čidlu TMP 36 a u jiných čidel se může lišit.

Nyní zapište a odlaďte následující kód, který obsahuje výše popsané vzorce:

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

Mezi jednotlivými měřeními je pauza 10 sekund. Tu si samozřejmě můžete upravit, dle vlastního přání.

Počítejte s tím, že po zapojení chvíli trvá, než se teplotní čidlo srovná na teplotu měřeného okolí. Zejména pokud jste jej před tím drželi delší dobu v ruce. První dva až tři výsledky doporučujeme ignorovat. Všimněte si, jak se teplota postupně bude ustalovat na určité hodnotě.

Zkuste teplotu porovnat s jiným teploměrem. Pokud se výsledky významně liší, zkuste ověřit, zda výstupní napětí vašeho micro:bitu je opravdu 3,18 V. Rovněž ověřte, zda vaše teplotní čidlo opravdu měří teplotu dle výše uvedeného vzorce.

Micro:bit obsahuje vestavěné teplotní čidlo měřící teplotu jeho procesoru. Jeho výsledky mohou být zejména při dlouhodobém měření, kdy se micro:bit ohřeje, vyšší. Program, který jej využívá by pak vypadal asi takto:

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

Zkuste porovnat teploty naměřené oběma způsoby.

Zkuste sestavit program, kde se budou střídat výsledky změřené oběma způsoby, případně, kde se po stisku tlačítek A či B zobrazí teplota měřená odpovídajícím způsobem.