

## 3 HUDBA

### Co se naučíte

- 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

Toto je verze pro Micro:bit V2. V této verzi již nepotřebujete připojovat externí reproduktory neboť tato verze Micro:bitu obsahuje buzzer a programy tak budou fungovat přímo bez připojování dalšího hardware.

Navíc tato verze obsahuje mikrofón, takže si můžeme vyzkoušet reakci na zvuk, např. Tlesknutí.

**POZOR** – doporučuji používat editor Thonny a programy pouze spouštět, ale nenahrávat. V této hodině reálně hrozí znečištění třídy hlukem z programů spuštěných ve smyčce. Takto spuštěný program lze kdykoliv přerušit stiskem tlačítka RESET na zadní straně Micro:bitu nebo odpojením Micro:bitu od kabelu.

Pokud použijete editor Mu nebo program nahrajete, pak se po uvolnění tlačítka RESET nebo připojení Micro:bitu opět spustí zvuk.

# 1.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
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

## 1. krok 5 minut

Rozdejte studentům micro:bity a kabely. Poučte studenty o tom, že při testování zvuku mají používat editor Thonny a programy pouze spouštět a nikoliv nahrávat na Micro:bit. Rovněž je poučte o tom, že programy přeruší stiskem tlačítka RESET.

## 2. krok 10 minut

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

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

Na řádce 2 je informace o přidání knihovny pro přehrání zvuku. Na řádce 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 zmáčkněte na Micro:bitu Reset.

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

## 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.

## 4. krok 15 minut

Na závěr hodiny vyzkoušíme ještě práci s mikrofonom. Zapište a odlaďte následující program (pouze pro editor Thonny):

```
from microbit import *
while True:
    sleep(1000)
    print(microphone.sound_level())
```

Po spuštění se rozsvítí ikonka mikrofonu a Thonny bude do dolní části po vteřině vypisovat hodnotu hluku v okolí. Hodnoty jsou 0 (ticho) až 255 (maximální hluk). Zabraňte studentské soutěži ve vytvoření co největší hodnoty.

Nyní když umíme odhadnout co vydává jak silný zvuk, můžeme nechat micro:bit reagovat na okolní zvuky např. takto:

```
from microbit import *
display.clear()
while True:
    if microphone.sound_level() > 8:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()
```

Pokud je úroveň zvuku větší než osm ukáže Micro:bit na displeji smajlík.

Microbit obsahuje dvě proměnné SoundEvent.LOUD a Sound.Event.QUIT. Jejich použití ukazuje následující příklad:

```
from microbit import *
display.clear()
microphone.set_threshold(SoundEvent.LOUD, 10)

while True:
    if microphone.current_event() == SoundEvent.LOUD:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()
```

Na třetím řádku nastavíme hodnotu proměnné SoundEvent. Každý zvuk s touto a vyšší hodnotou je nyní brán jako hlasitý.

## 2.PRACOVNÍ LIST III-1

Naučíte se k micro:bitu připojit sluchátka a přehrát na nich melodii.

### 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 micro USB koncovkou
- Micro:bit

POZOR: Tento text platí pouze pro Micro:bit V2. Pro verzi jedna použijte pracovní list s připojováním externího reproduktoru.

### A jděte na to ...

Nahrajte do micro:bitu následující program:

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

Příkaz na řádce 2 zavádí knihovnu pro práci se zvukem a na řádce 3 se přehraje připravený zvuk. Seznam všech připravených melodií vám poskytne vyučující.

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 s jinými obrázky a melodiemi.

Nyní si vyzkoušíme práci s mikrofonom. Zapište a odlaďte následující program (pouze pro editor Thonny):

```
from microbit import *
while True:
    sleep(1000)
    print(microphone.sound_level())
```

Po spuštění se rozsvítí ikonka mikrofону a Thonny bude do své dolní části po vteřině vypisovat hodnotu hluku v okolí. Hodnoty jsou 0 (ticho) až 255 (maximální hluk). Všimněte si, že mikrofon je poměrně citlivý.

Nyní když umíme odhadnout co vydává jak silný zvuk, můžeme nechat micro:bit reagovat na okolní zvuky např. takto:

```
from microbit import *
display.clear()
while True:
    if microphone.sound_level() > 8:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()
```

Pokud je úroveň zvuku větší než osm ukáže Micro:bit na displeji smajlík. Experimentujte se změnou úrovně.

Microbit obsahuje dvě proměnné `SoundEvent.LOUD` a `Sound.Event.QUIT`. Jejich použití ukazuje následující příklad:

```
from microbit import *
display.clear()
microphone.set_threshold(SoundEvent.LOUD, 10)

while True:
    if microphone.current_event() == SoundEvent.LOUD:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()
```

Na třetím řádku nastavíme hodnotu proměnné `SoundEvent`. Každý zvuk s touto a vyšší hodnotou je nyní brán jako hlasitý.

Úkol: Vytvořte program, který zajistí, že když je v okolí klid bude se Micro:bit usmívat, pokud však bude velký hluk, bude Micro:bit mračit. Bude-li úroveň mezi bude displej prázdný. Stanovte si sami úrovně hlasitosti.

### 3.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
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

#### 1. krok 10 minut

Rozdejte studentům micro:bity. 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 je vynechat včetně čárky. 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 určitě nepoužívejte české znaky.

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

#### 2. krok 20 minut

Pak je nechte napsat a odladit následující program, který přehraje melodii ovčáci čtveráci. V tomto případě je možné pro zmenšení počtu chyb tento program vhodným způsobem studentům vysdílet. 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. 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 obou micro:bitech. Později se studenti naučí též synchronizaci pomocí rádia, která by byla vhodnější, ale zatím jí nemůžete použít.

## PRACOVNÍ LIST III-2

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

### Co se naučíte

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

### Co budete potřebovat

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

### A jděte na to ...

Napište a odlaďte následující program:

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

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 je vynechat včetně čárky. (Defaultní hodnota je 72, ale 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 tak nutno použít anglickou transkripci. Např. „Josef“ je třeba napsat jako „Yoseph“ atd. A samozřejmě nelze použít české znaky.

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



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). Příkaz `music.play(nota)` pak daný záznam přehraje.

Otázka: Co je za strukturu `nota`?

Zkuste si naprogramovat vlastní melodii nebo nějakou známou skladbu.

# PRŮVODCE TEORIÍ

## Připojení audio výstupu

Micro:bit V2 má nově buzzer, který umí vybzučet jednoduché melodie a mikrofon, který umožňuje reagovat na hladinu zvuku v okolí, případně na akustické signály.

## Přehrávání připravených melodií

MicroPython obsahuje asi dvacet předem připravených melodií, jejichž seznam najdete v dokumentaci. Ukázka použití je v následujícím příkladu:

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

Všimněte si, že na řádce 2 je nutné zavést knihovnu pro přehrávání hudby. Samotný příkaz pro přehrání melodie je pak na řádce 3.

Seznam všech připravených melodií naleznete v příloze B nebo na stránkách dokumentace MicroPythonu pro micro:bit.

Připravené melodie lze dobře kombinovat s připravenými obrázky, jak ukazuje další příklad:

```
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()
```

Význam jednotlivých příkazů už by vám měl být jasný a proto neuvádíme žádný další popis.

## Mikrofon

Zapište a odlaďte následující program (pouze pro editor Thonny):

```
from microbit import *
while True:
    sleep(1000)
    print(microphone.sound_level())
```

Po spuštění se rozsvítí ikonka mikrofonu a Thonny bude do dolní části po vteřině vypisovat hodnotu hluku v okolí. Hodnoty jsou 0 (ticho) až 255 (maximální hluk).

Nyní když umíte odhadnout co vydává jak silný zvuk, můžete nechat micro:bit reagovat na okolní zvuky např. takto:

```

from microbit import *
display.clear()
while True:
    if microphone.sound_level() > 8:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()

```

Pokud je úroveň zvuku větší než osm ukáže Micro:bit na displeji smajlík.

Microbit obsahuje dvě proměnné `SoundEvent.LOUD` a `Sound.Event.QUIT`. Jejich použití ilustruje následující příklad:

```

from microbit import *
display.clear()
microphone.set_threshold(SoundEvent.LOUD, 10)

while True:
    if microphone.current_event() == SoundEvent.LOUD:
        display.show(Image.HAPPY)
        sleep(500)
        display.clear()

```

Na třetím řádku nastavíme hodnotu proměnné `SoundEvent`. Každý zvuk s touto a vyšší hodnotou je nyní brán jako hlasitý.

## Micro:bit mluví

Microbit umí i mluvit. Naneštěstí pro nás pouze anglicky. Knihovna pro mluvení je navíc zatím označena jako vývojová, takže se můžete potkat s chybami. Mluvení je velmi jednoduché:

```

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

```

Na řádku 2 se zavádí knihovna pro hovor a na řádku 3 je zadán příkaz pro mluvení. Zde micro:bit pozdraví. Parametr `speed=100` je nepovinný. 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. Nezapomeňte pro slova použít anglickou transkripci např. "Yoseph" pro Josef.

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.

## Přehrání not

Micro:bit dovede přehrát noty. Následující program přehraje melodii „Ovčáci, čtveráci“. Zázpis programu trochu připomíná vytváření animovaných obrázků.

```
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)
```

Struktura seznam (list) nota je vlastně zápis jednotlivých tónů. Např. 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.

# PŘÍLOHA – SEZNAM PŘIPRAVENÝCH MELODIÍ

- `music.DADADADUM`
- `music.ENTERTAINER`
- `music.PRELUDE`
- `music.ODE`
- `music.NYAN`
- `music.RINGTONE`
- `music.FUNK`
- `music.BLUES`
- `music.BIRTHDAY`
- `music.WEDDING`
- `music.FUNERAL`
- `music.PUNCHLINE`
- `music.PYTHON`
- `music.BADDY`
- `music.CHASE`
- `music.BA_DING`
- `music.WAWAWAWAA`
- `music.JUMP_UP`
- `music.JUMP_DOWN`
- `music.POWER_UP`
- `music.POWER_DOWN`