

# 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 řádku 2 je nutné zavést knihovnu pro přehrávání hudby. Samotný příkaz pro přehrání melodie je pak na řádku 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“. Zápis 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`