

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

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` 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 stisknuté 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 řádce 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 piezzobuzzerů.

Doporučujeme, aby studenti měli sluchátka ať se vzájemně nepřehlušují rámcem. 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.

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, odlad'te 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 vrátí informaci, zda tlačítko bylo stisknuté od minulé kontroly nebo o d začátku programu, pokud jeho stisknutí nebylo dosud kontrolováno.

Nyní si vyzkoušíte práci s oběma tlačítky současně. Odlad'te 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 programu nahraďte 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.

PRŮVODCE TEORIÍ

Micro:bit obsahuje celkem tři tlačítka. Tlačítko, umístěné na zadní straně mezi vstupy pro USB kabel a napájecí kabel, je tlačítko reset. To nás teď nebude zajímat. Na přední straně jsou umístěna dvě programovatelná tlačítka, A a B, jimž se budeme v této kapitole věnovat.

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()
```

Práce s tlačítky je ukázána na řádcích 3 a 5. Jedná se vlastně o dotaz, zda je tlačítko stisknuté. Micropython má, jak vidíte připraveny dvě proměnné `button_a` a `button_b`. Funkce `button_a.is_pressed()` vrátí 1, pokud je tlačítko stisknuté jinak vrátí 0. Existuje ještě funkce `button_a.was_pressed()`, která testuje, zda tlačítko bylo stisknuté od minulé kontroly nebo od zapnutí micro:bitu. Obdobné funkce jsou i pro tlačítko b.

Chcete-li testovat současný stisk obou tlačítek použijte následující programové konstrukci:

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

Mezi oběma testovacími funkcemi na řádku 3 je použita logická spojka `and`, která znamená, že celkově podmínka platí pouze pokud platí obě dílčí podmínky – jsou stisknuta obě tlačítka současně.

Naopak, pokud testujete, zda je stisklé libovolné tlačítko (A nebo B), použijte následující konstrukci se spojkou `or` (nebo):

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

Kromě uvedených funkcí `is_pressed` a `was_pressed`, je pro objekty `button_a` a `button_b` definována ještě funkce `get_presses()`. Tato funkce zjistí počet stisknutí tlačítka od posledního testování a nastaví jej na nulu.

Následující příklad vyčká po zapnutí (nebo stisku reset) micro:bitu deset sekund a pak zobrazí počet stisků tlačítka A během této doby (od zapnutí nebo resetu):

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