

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. Pozor nefungují všechny kabely. Pokud budete používat jiné než koupené spolu s micro:bitem, je nutné je předem vyzkoušet.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro studenty

1. krok 5 minut

Vysvětlíte, že micro:bit obsahuje 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 ví 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 tímto 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()
```

Pozor **čísla nejsou součástí programu**, slouží pouze pro možnost odkazovat se na konkrétní řádek.

Příkazy pro dotaz zda je tlačítko stisknuto jsou na řádcích 3 a 5. Nechte studenty přijít na jejich význam – test stisku tlačítka. Funkce (metoda) vrací 1 (stisknuto) nebo 0 (nestisknuto). 1 znamená splněná podmínka (True), 0 nesplněná (False). Existuje i funkce `button_a.was_pressed()`, která testuje, zda tlačítko bylo stisklé od minulého testování nebo od začátku programu.

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

3. krok 10 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.

Program nepatrně změňte (pouze na řádce 3):

```
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 5 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())
```

5. krok 15 minut – dotekové tlačítko

Piny 0, 1 a 2 na přední straně můžeme použít jako dotekové tlačítko. Použití je velmi jednoduché, odkazujeme se na ně jako na `pin0`, `pin1` a `pin2`. Viz následující příklad:

```
from microbit import *
while (True):
    if pin0.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
```

Je nutné stisknout současně odpovídající pin a GND. Je třeba stisk z obou stran a je tedy nutné použít obě ruce. Při správném stisku se zobrazí na displeji A, jinak se zobrazí N. Nechte studenty vyzkoušet.

Dotekové logo – pouze Micro:bit v.2

Micro:bit ve verzi 2 má místo loga na přední straně dotekové tlačítka. Jeho použití je rovněž velmi jednoduché, odkazujeme se na něj jako na pin_logo. Viz následující příklad, který z předchozího vznikne pouze úpravou jména pinu:

```
from microbit import *
while (True):
    if pin_logo.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
    if pin0.is_touched():
```

V tomto případě stačí po logu jenom přejet jedním prstem.

POZOR: Tento příklad opravdu funguje pouze na Micro:bitu v.2

Pokud máte k dispozici Micro:bit v.2 je možné tuto hodinu rozdělit na dvě hodiny a nechat studenty samostatně řešit následující příklad. Toto rozdělení je samozřejmě možné i u Micro:bitu v.1. Hodinu bychom doporučovali v takovémto případě rozdělit před krokem 5.

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. V případě Micro:bitu v.2 lze pro konec použít stisk loga. 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 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 rámušem. Vy si naopak připravte repráčky ať vše můžete dobře demonstrovat.

POZOR: Pokud máte Micro:bit v.2 pak již není potřeba ani sluchátka, ani repráček. Důrazně doporučujeme pro testování zvuku editor Thonny, který defaultně nenahrává program na Micro:bit a současně umožňuje běžící program zastavit a ukončit potencionálně nepříjemný zvuk.

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
- Vnořené funkce

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel 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 запиšte, 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 **čísla nejsou součástí programu**, slouží pouze pro možnost odkazovat se na konkrétní řádek.

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 stisklé od začátku programu nebo od minulé kontroly.

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 program dělá?

Jaký je význam logické spojky **and** na řádce 3?

Program nepatrně změňte na řádce 3:

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

Jaká je změna ve funkci programu?

Jaký je význam logické spojky **or**?

Nyní запиšte a odlad'te 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.

Piny 0, 1 a 2 na přední straně můžeme použít jako dotekové tlačítko. Použití je velmi jednoduché, vyzkoušejte následující příklad:

```
from microbit import *
while (True):
    if pin0.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
```

Je nutné stisknout současně odpovídající pin 0 a GND. Je třeba stisk z obou stran a je tedy nutné použít obě ruce. Při správném stisku se zobrazí na displeji A, jinak se zobrazí N. Obdobně lze použít piny 1 a 2.

POZOR: Následující příklad opravdu funguje pouze na Micro:bitu v.2

Micro:bit ve verzi 2 má místo loga na přední straně dotekové tlačítko. Jeho použití je rovněž velmi jednoduché viz následující příklad, který z předchozího příkladu vznikne pouze malou úpravou:

```
from microbit import *
while (True):
    if pin_logo.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
    if pin0.is_touched():
```

V tomto případě stačí po logu jenom přejet jedním prstem.

Neřešený závěrečný příklad: Naprogramujte postřehovou hru. Na Micro:bitu se bude střídavě 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 z kapitoly 1.

PRŮVODCE TEORIÍ

Micro:bit obsahuje celkem tři tlačítka. Tlačítko umístěné na zadní straně mezi vstupy pro USB kabel a kabel napájení je tlačítko *reset* a dále vás nebude zajímat. Na přední straně jsou umístěna dvě programovatelná tlačítka A a B. Jejich programováním a využitím se bude zabývat tato kapitola.

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 tlačítko je zmáčknuté. Micropython má jak vidíte připraveny dvě proměnné `button_a` a `button_b`. Funkce `button_a.is_pressed()` vrací 1, pokud je tlačítko stisknuté jinak vrací 0. Existuje ještě funkce `button_a.was_pressed()`, která testuje zda tlačítko bylo stisknuté od minulé kontroly nebo od zapnutí micro:bitu.

Chcete-li testovat současný stisk obou tlačítek použijte následující konstrukce:

```
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())
```

Dotekové tlačítko

Piny 0, 1 a 2 na přední straně můžeme použít jako dotekové tlačítko. Použití je velmi jednoduché, odkazujeme se na ně jako na pin0, pin1 a pin2. Viz následující příklad:

```
from microbit import *
while (True):
    if pin_logo.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
```

Je nutné stisknout současně odpovídající pin a GND. Je třeba stisk z obou stran a je tedy nutné použít obě ruce. Při správném stisku se zobrazí na displeji A, jinak se zobrazí N.

Dotekové logo – pouze Micro:bit v.2

Micro:bit ve verzi 2 má místo loga na přední straně dotekové tlačítko. Jeho použití je rovněž velmi jednoduché, odkazujeme se na něj jako na pin_logo. Viz následující příklad, který z předchozího vznikne pouze úpravou jména pinu:

```
from microbit import *
while (True):
    if pin_logo.is_touched():
        display.show("A")
    else:
        display.show("N")
    sleep(100)
    if pin0.is_touched():
```

V tomto případě stačí po logu jenom přejet jedním prstem.

POZOR: Tento příklad opravdu funguje pouze na Micro:bitu v.2