

1. VÝPIS NA LED DISPLAY

Co se naučíte

- Ovládat matici 5x5 LED diod na micro:bitu
- Zobrazit běžící text nebo jeden statický znak
- Zobrazit přednastavený obrázek
- Vytvořit jednoduchou animaci

Co budete potřebovat

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

Časová náročnost

3 až 5 vyučovacích hodin 45 minut

PRŮVODCE HODINOU I-1

Studenti se seznámí s *micro:bitem* a textovým editorem *mu*. Naprogramují základní úlohu – běžící text „Ahoj světe!“.

Co bude v této hodině potřeba:

- PC se editorem *mu*. Lze použít libovolný operační systém. Editor stáhnete na stránce <https://codewith.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 10 minut

Vysvětlíte studentům, co se naučí v tomto předmětu:

- Poznájí výukovou platformu *micro:bit* a naučí se jí ovládat
- Naučí se některým principům z elektronických obvodů
- Připomenou si anebo se seznámí s jazykem Python ve verzi MicroPython

Rozdejte studentům pracovní listy a *micro:bity*

2. krok 10 minut

Popište studentům *micro:bit*. Řekněte něco o jeho historii a možnostech a použití. Nechte studenty aby si jej prohlédli. Promítejte během výkladu *micro:bit* na projektor z prezentace. Zmiňte jeho následující vlastnosti:

- matice 5x5 diod
- dvě programovatelná tlačítka zepředu, tlačítko reset vzadu
- vstupy – micro USB a napájecí konektor
- akcelerátor, magnetometr
- možnost měřit intenzitu osvětlení a teplotu (ale obojí ne zcela přesně)
- připojení k mobilu přes bluetooth (bohužel v MicroPythonu nepoužitelné)
- možnost vzájemné komunikace
- 17 **GPIO** (general-purpose input/output) pinů, dále piny 3V a GND. Tři piny jsou větší (označené 0, 1, 2) pro snadné připojení vodičů s krokodýlky. Ostatní piny pro plnohodnotné použití vyžadují speciální základnu.

3. krok 10 minut

Představte jazyk *Python* a *MicroPython* – podmnožinu *Pythonu*, obsahující navíc knihovny pro práci s micro:bitem. Představte editor *mu* a jeho možnosti. Při výkladu promítejte editor na projektor (např. z prezentace). Proberte krátce význam tlačítek.

- New, Load, Save – Nový program, nahrání programu, uložení (při prvním uložení se ptá na jméno programu)
- Flash – Nahrání programu 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
- Zoom in, zoom out – velikost písma
- Theme – světlý text na tmavém pozadí a naopak
- Check – kontrola syntaxe. Většina chyb se projeví až při spuštění. Opětovné stisknutí vypne zobrazení chyb a následující jej opět zapne.
- Help – odkaz na stránky s nápovědou
- Quit – konec

4. krok 15 minut

Studenti napíší a spustí první program – ahoj_sвете.py. Řekněte studentům, aby si spustili editor *mu* a připojili micro:bit USB kabelem. Do editoru napíší dva řádky kódu:

```
1. from microbit import *  
2. display.scroll("Ahoj světe")
```

Promítněte studentům kód na projektor a vysvětlete význam obou řádků.

Vysvětlete studentům, že **nelze používat české znaky** (a to ani v textech), protože je micro:bit neumí zobrazit.

Nechte studenty zkontrolovat syntaxi a řešte s nimi chyby.

První řádek budou možná již mít v editoru *mu* přednastavený. Nechte studenty nahrát program na micro:bit a řešte s nimi případné chyby.

- Program nelze nahrát – zkontrolujte zda je micro:bit připojený, zkuste jiný kabel, USB port, micro:bit, počítač.
- Microbit píše něco jiného – informaci o chybě. Podívejte se do editoru *mu*. Na řádku kde je chyba uvidíte na začátku červenou šipku. Po kliknutí na ní se dozvíte o jaký problém se jedná
- Program musí končit odřádkováním a po něm následovat prázdný řádek

Pokud zbude čas, zkuste ještě program s nekonečnou smyčkou ahoj_sвете2.py, jinak jej ponechte na začátek další hodiny.

PRACOVNÍ LIST I-1

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

Co se naučíte

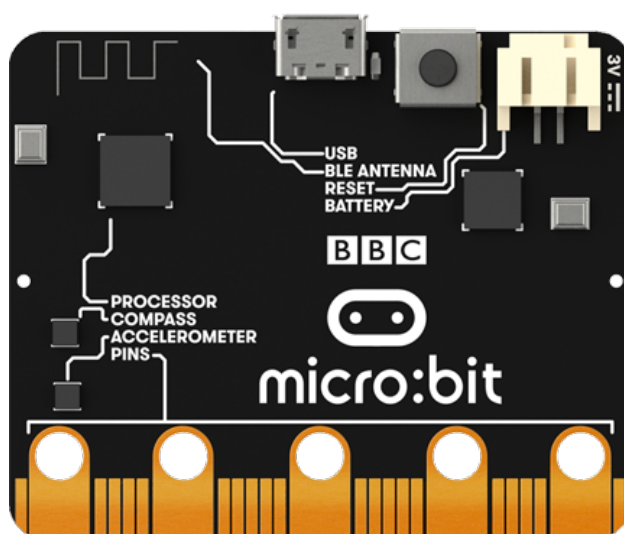
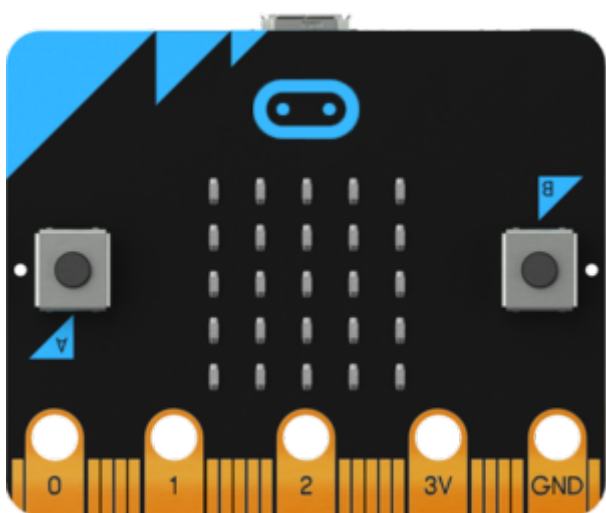
- Základ práce s *micro:bit*em
- Práci s editorem *mu*
- Odladění prvního programu

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*.



Na přední straně (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 s velkými konektory označené 0, 1 a 2 a vedle nich výstup na 3 V a zem označená GND.

Micro:bit má dva vstupy *micro USB* a napájecí vstup. Napájení 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.

Nyní spustě program *mu*. Význam tlačítek nahoře je:

- New, Load, Save – Nový program, nahrání programu, uložení (při prvním se ptá na jméno)
- Flash – Nahrání programu 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
- 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 :

```
1. from microbit import *
2. display.scroll("Ahoj svete")
3.
```

Na konci řádku 3 musí být odřádkování (program končí prázdným řádkem) a na řádku 4 už nesmí být žádný text (ani mezery). Připojte micro:bit a vyčkejte cca. 10 sekund. Program uložte a stiskněte tlačítko Flash. Pokud je vše v pořádku program se nahraje do micro:bitu (po dobu nahrávání bliká žlutá dioda na zadní straně). Nyní by měl po micro:bitu proběhnout text Ahoj svete.

Možné chyby

Program nelze nahrát – zkontrolujte zda je micro:bit připojený, zkuste v tomto pořadí jiný kabel, jiný USB port , jiný micro:bit, jiný počítač.

Microbit píše něco jiného – informaci o chybě. Podívejte se do editoru mu. Na řádku kde je chyba uvidíte na začátku červenou šipku. Po kliknutí na ní se dozvíte o jaký problém se jedná

Program musí končit odřádkováním a po něm následovat prázdný řádek

Další program

Máte-li hotovo můžete vyzkoušet ještě tento program:

```
4. from microbit import *
5.
6. while True:
7.     display.scroll("Ahoj svete")
8.     sleep(1000)
```

Tento program v nekonečné smyčce vypisuje Ahoj svete, vyčká 1 vteřinu a znovu. Zápis nekonečné smyčky je na řádku 3. Na řádku 5 je příkaz, že program má čekat 1000 tisícín vteřiny – tedy jednu vteřinu.

Pozor na chyby:

- Příkazy na řádku 4 a 5 musí být odsazeny zleva přesně o čtyři mezery. Mu vám to bude už nabízet. Nelze použít jiný počet mezer nebo tabelátor jak bývá obvyklé v jiných verzích Pythonu.
- Na konci řádku 3 je dvojtečka, na to začátečníci často zapomínají
- Čísla nejsou součástí kódu, jsou zde uvedena jen pro lepší orientaci v programu.
- Volný řádek 2 není povinný, je jen pro přehlednost.

Důležité webové adresy

Domácí stránka Micro:bitu

<https://microbit.org/>

Dokumentace k MicroPythonu:

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

Editor Mu:

<https://codewith.mu/>

PRŮVODCE HODINOU I-2

V této hodině nejprve rozšíříme příklad Ahoj_sвете z minulé hodiny. Později přidáme dva další příklady (jeden z nich ve dvou modifikacích). Na těchto příkladech si současně připomeneme nebo se naučíme psaní cyklů v Pythonu.

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

Řešte úlohu s nekonečným výpisem textu „Ahoj světe“. Vysvětlíte studentům význam cyklu while True:

Obraťte pozornost na správnou syntaxi.

- True musí být s velkým T.
- Na konci řádku se zápisem cyklu musí být znak dvojtečky. Na ten se často zapomíná.
- Odsazení těla cyklu musí být o právě čtyři mezery. Editor mu vám pomůže automatickým odsazením dalších řádků.
- Na konci musí být odřádkování a na posledním řádku nesmí být nic (ani mezery).

```
1. from microbit import *
2.
3. while True:
4.     display.scroll("Ahoj světe")
5.     sleep(1000)
```

2. krok – 25 minut

Napište dvěma různými způsoby program, který vypíše čísla od 1 do 10 a pak skončí.

Použijete postupně cykly for a while.

Zápis s cyklem for:

```
1. from microbit import *
2.
3. for i in range(1, 11):
4.     display.scroll(str(i))
```

Vysvětlíte syntaxi programu. Jedná se o *cyklus s pevným počtem opakování*. Je třeba vysvětlit, že zápis range (1, 11) znamená v Pythonu rozsah od 1 do 10. Je to vždy o jednu méně než je mez vpravo – **častý zdroj chyb**. Dále je třeba říci, že za čárkou musí být mezera. Zkuste studentům říci některé příklady zápisu range:

- `range(3, 1)` – cyklus se neprovede (3 je větší než 1)
- `range(1, 4, 2)` – cyklus se provede pro 1 a 3. Iterace je po dvou.
- `range(3, 1, -1)` – cyklus se provede pro 3 a 2.

Nechte studenty případně vyzkoušet.

Zeptejte se studentů

Proč je uvnitř `display.scroll()` příkaz `str()`?

Protože `display.scroll()` umí vypsát pouze řetězec, použijeme příkaz `str()`, který převede celé číslo na řetězec.

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

Řetězec je posloupnost znaků anglické abecedy, čísel a dalších znaků. Přesněji definováno, jedná se o znaky s ASCII kódem 32 až 127. Definici celého čísla známe z matematiky. Zcela přesně je to číslo bez desetinné čárky z intervalu $\leftarrow 2147483648, 2147483647 \rightarrow$.

Zápis s cyklem while:

Jedná se o cyklus s neurčitým počtem opakování.

Pohovořte o dané syntaxi. Za příkazem `while` následuje podmínka a cyklus (odsazené řádky za dvojtečkou) se provádí tak dlouho dokud podmínka platí.

```
1. from microbit import *
2.
3. i = 1
4. while (i < 11):
5.     display.scroll(str(i))
6.     i = i + 1
```

Všimněte si práce s proměnnými – definice proměnné na řádce 3 a změny její hodnoty na řádce 6
Pozor kolem = a matematický operátorů **musí být mezery**.

Pozor na mezery. Např. V zápisu: `i = i + 1` musí být mezera jak kolem znaku = tak okolo znaku +.

Zkuste se studentů zeptat, zda je jim bližší zápis s `for` nebo s `while`?

Vysvětlete pojem negace podmínky – `not(podmínka)`

Je totéž `while not(i > 11):` ? Není správně by bylo: `while not(i >= 11):`

3. krok – 10 minut

Ukázka dalších funkcí pro objekt `display`:

```
1. from microbit import *
2. display.show("X")
3. sleep(1000)
4. display.clear()
```


Příklad zobrazí znak X pomocí `display.show()` po dobu jedné sekundy a pak smaže displej pomocí `display.clear()`.

`display.show()` zobrazí řetězec nebo číslo znak po znaku. Prodlevu mezi změnami lze nastavit (v milisekundách pomocí parametru. Např. `display.show("Ahoj svete", 1000)` .

Poslední znak zůstane svítit na displeji. Pokud je parametrem jen jeden znak nebo jednociferné číslo, pak toto zůstane svítit tak dlouho než zobrazíte něco jiného nebo smažete obrazovku pomocí `display.clear()`.

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.

```
1. from microbit import *
2.
3. while True:
4.     display.scroll("Ahoj svete")
5.     sleep(1000)
```

Jedná se o nekonečnou smyčku.

Pozor na syntaxi:

- True musí být s velkým T
- Na konci zápisu cyklu je dvojtečka
- Odsazení musí být o přesně čtyři znaky

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

```
1. from microbit import *
2. for i in range(1, 11):
3.     display.scroll(str(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. Přeložte program a vyzkoušejte.

Otázky:

Proč je uvnitř `display.scroll()` příkaz `str()`?

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

Nyní totéž pomocí cyklu `while`:

```
4. from microbit import *
5.
6. i = 1
7. while (i < 11):
8.     display.scroll(str(i))
9.     i = i + 1
```

Otázky:

Co znamená negace?

Je totéž `(i > 11)` a `not(i < 11)`?

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

PRŮVODCE HODINOU I-3

Studenti se seznámí s grafikou na displeji micro:bitu. Vyzkouší si jak zobrazení připravených obrázků, tak tvorbu obrázků vlastních.

Co bude v této hodině potřeba:

- PC se editorem mu.
- Micro:bit s USB kabelem.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro studenty

1. krok 20 minut

Vyzkoušejte zobrazení předpřipravených obrázků. Micro:bit v této ukázce střídá náladu:

```
1. from microbit import *
2. display.show(Image.SAD)
3. sleep(1000)
4. display.show(Image.SMILE)
5. sleep(1000)
6. display.show(Image.HAPPY)
7. sleep(1000)
8. display.clear()
```

Zvažte jakým způsobem studentům dát seznam obrázků. Zda jim poskytnout odkaz na web MicroPythonu nebo seznam (v příloze této kapitoly) vytisknout.

Následuje další ukázka, která simuluje 100 úderů srdce:

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

Zeptejte se studentů:

Proč je použit zrovna sleep(400)?

Celý cyklus takto trvá 0,8 sekundy, což vede na frekvenci 75 tepů za minutu.

2. krok 25 minut

Napište a odlaďte následující program:

```
1. from microbit import *
2. raketa = Image("00900:"
3.               "05550:"
4.               "05550:"
5.               "09990:"
6.               "90909:")
7. display.show(raketa)
```

Pozor na syntaxi obrázku:

- Každý řádek kódu je řádek displeje
- Každý řádek je uvozen apostrofem a uvnitř končí dvojtečkou
- Čísla od 0 do 9 znamenají intenzitu světla (0 – nesvítí, 9 – svítí naplno)

Poskytněte studentům prostor pro sestavení vlastního obrázku.

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-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
- Nastavení intenzity konkrétní diody

Co budete potřebovat

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

A jděte na to ...

Zapište a odlad'te následující kód:

```
1. from microbit import *
2. display.show(Image.SAD)
3. sleep(1000)
4. display.show(Image.SMILE)
5. sleep(1000)
6. display.show(Image.HAPPY)
7. sleep(1000)
8. display.clear()
```

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

Zkuste ještě následující příklad simulující 100 úderů srdce:

```
9. from microbit import *
10. for i in range(1, 100):
11.     display.show(Image.HEART)
12.     sleep(400)
13.     display.show(Image.HEART_SMALL)
14.     sleep(400)
15. display.clear()
```

Otázky:

Přemýšlejte, proč je použita zrovna konstanta `sleep(400)`?

Nyní zkuste následující příklad, který má vytvořit na displeji obrázek rakety:

```
1. from microbit import *
2. raketa = Image("00900:"
3.               "05550:"
4.               "05550:"
5.               "09990:"
6.               "90909:")
7. display.show(raketa)
```

Pozor na syntaxi obrázku:

- Každý řádek kódu je řádek displeje
- Každý řádek je uvozen apostrofem a uvnitř končí dvojtečkou
- Čísla od 0 do 9 znamenají intenzitu světla (0 – nesvítí, 9 – svítí naplno)

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

PRŮVODCE HODINOU I-4 (I-5)

Studenti se seznámí s pokročilejší grafikou na micro:bitu. Naučí se tvorbě animace a adresaci konkrétní diody displeje.

Jakým způsobem a zda vůbec učit tuto část ponecháváme na učitelích. Je možné tuto část vypustit buď zcela nebo první či druhou část. Nebo je naopak možné tuto kapitolu rozdělit do dvou samostatných hodin. Pokud učíte dvouhodinovky, je možné první část připojit k hodině III a ke druhé části v následující dvouhodinovce přidat opakování celé této části. Rozhodně by se s ní měl nejprve učitel dobře seznámit a rozhodnout dle úrovně svých studentů.

První program v této kapitole je poměrně rozsáhlý. Zvažte proto možnost jeho zdrojový kód tentokrát žákům poskytnout, aby jej nemuseli opisovat. Pokud naopak je necháte kód opisovat, např. z důvodu procvičení ladění programu, pak počítejte s nutností rozdělit kapitolu do dvou hodin. Z výše uvedených důvodů tentokrát neuvádíme časovou náročnost jednotlivých částí.

Co bude v této hodině potřeba:

PC se editorem mu.

Micro:bit s USB kabelem.

Pokud je k dispozici, tak dataprojektor

Prezentaci k této lekce

Pracovní listy pro studenty

Volitelně: Připravený zdrojový kód programu z 1. kroku, umístěný tak, aby k němu měli žáci přístup.

1. krok

Naeditujte následující kód a nahrajte jej do micro:bitu:

```
8. from microbit import *
9. raketa1 = Image("00900:"
10.                "05550:"
11.                "05550:"
12.                "09990:"
13.                "90909:")
14. raketa2 = Image("00900:"
15.                "05550:"
16.                "05550:"
17.                "09990:"
18.                "99999:")
19. raketa3 = Image("05550:"
20.                "05550:"
21.                "09990:"
22.                "99999:"
23.                "00000:")
24. raketa4 = Image("09990:"
25.                "99999:"
26.                "00000:"
27.                "00000:"
28.                "00000:")
29. raketa5 = Image("99999:"
30.                "00000:"
31.                "00000:"
32.                "00000:"
33.                "00000:")
34. raketa6 = Image("00000:"
35.                "00000:"
36.                "00000:"
37.                "00000:"
38.                "00000:")
39. raketa = [raketa1, raketa2, raketa3, raketa4, raketa5,
40.            raketa6]
40. display.show(raketa, delay=500)
```

Řekněte studentům:

- Jedná se vlastně o postupné zobrazení obrázků `raketa1` až `raketa6` po půl sekundě
- Struktura `raketa` se nazývá **list (seznam)** – jedná se o uspořádanou n-tici, u které záleží na pořadí a umožňuje opakovaný výskyt jednotlivých prvků

Úkol pro samostatnou práci:

Je možné vypustit obrázek `raketa6`? Pokud ano, je nutná úprava programu?

Ano je to možné. K programu je pak nutné doplnit tyto dva řádky:

```
41. sleep(500)
42. display.clear()
```

Máte-li čas, nechte studenty vytvořit vlastní animaci.

2. krok

Pro adresaci konkrétního bodu displeje slouží příkaz `display.set_pixel(X, Y, intenzita)` s následujícím významem:

- X – sloupec, zleva doprava od 0 do 4
- Y – řádek shora dolů od 0 do 4
- intenzita – jas diody, 0 vypnutá, 9 zapnutá naplno.

Bod vlevo nahoře je tedy 0, 0, bod vpravo dole 4, 4.

V následujícím programu je použit **generátor náhodných čísel**. Pro jeho použití je nutno zavést knihovnu `import random` a pak je možné použít funkce `random.randint(A, B)`, která vrací náhodné číslo z uzavřeného intervalu A,B.

```
1. from microbit import *
2. import random
3. while True:
4.     x = random.randint(0, 4)
5.     y = random.randint(0, 4)
6.     intenzita = random.randint(0, 9)
7.     display.set_pixel(x, y, intenzita)
8.     sleep(10)
```

Program v nekonečném cyklu načítá náhodné souřadnice a intenzitu a s danými parametry rozsvěcí diodu, celkový dojem trochu připomíná hvězdy na noční obloze. Časová prodleva je přidána, aby nedocházelo k příliš rychlému blikání.

Zeptejte se studentů:

Jak pracuje generátor náhodných čísel?

Náhodné číslo je generováno např. na základě času od zapnutí a teploty okolního prostoru.

Tato dvě čísla se mohou sečíst a dělit nějakým prvočíslem a pak vzít číslo na konkrétní pozici jako výsledek.

Jedná se o analogové či o digitální zobrazení?

Jedná se o diskretizaci analogového zobrazení – výsledek může nabývat více než dvou hodnot, ale omezený počet (10).

Program nyní upravte:

```
1. from microbit import *
2. import random
3.
4. while True:
5.     x = random.randint(0, 4)
6.     y = random.randint(0, 4)
7.     if (display.get_pixel(x, y)):
8.         display.set_pixel(x, y, 0)
9.     else:
10.         display.set_pixel(x, y, 9)
11.     sleep(10)
```

Řekněte studentům:

- Jedná se de facto o digitalizaci, neboť v této úloze jsou použity pouze dvě úrovně rozsvícení diody 0 a 9.
- Funkce `display.get_pixel()` zjistí aktuální úroveň světla diody na dané souřadnici. Pokud vrátí hodnotu 0, podmínka není splněna. Následně je nastavena opačná intenzita.
- Pozor na dvojí úroveň odsazení. Ve druhé úrovni musí být 8 znaků (násobek 4).

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 o požadované intenzitě.

Co se naučíte

- Vytvoření animace
- Poznáte datovou strukturu list (seznam)
- Rozsvítit konkrétní diodu s požadovanou intenzitou
- Práci s generátorem náhodných čísel
- Zjištění intenzity 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 otevřete dle pokynů vyučujícího):

```
1. from microbit import *
2. raketa1 = Image("00900:"
3.               "05550:"
4.               "05550:"
5.               "09990:"
6.               "90909:")
7. raketa2 = Image("00900:"
8.               "05550:"
9.               "05550:"
10.              "09990:"
11.              "99999:")
12. raketa3 = Image("05550:"
13.               "05550:"
14.               "09990:"
15.               "99999:"
16.               "00000:")
17. raketa4 = Image("09990:"
18.               "99999:"
19.               "00000:"
20.               "00000:"
21.               "00000:")
22. raketa5 = Image("99999:"
23.               "00000:"
24.               "00000:"
25.               "00000:"
26.               "00000:")
27. raketa6 = Image("00000:"
28.               "00000:"
29.               "00000:"
30.               "00000:")
```

```
31.             "00000:")
32.raketa = [raketa1, raketa2, raketa3, raketa4, raketa5,
            raketa6]
33.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, je nutná úprava programu?

Zkuste si vytvořit vlastní animaci.

Nyní zkuste napsat a odladit následující program, který náhodně rozsvěcí diody s různou intenzitou a simuluje tak hvězdnou oblohu:

```
1. from microbit import *
2. import random
3. while True:
4.     x = random.randint(0, 4)
5.     y = random.randint(0, 4)
6.     intenzita = random.randint(0, 9)
7.     display.set_pixel(x, y, intenzita)
8.     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 intenzitu. Intenzita je celé číslo z uzavřeného intervalu 0,9. 0 – nesvítí, 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?

Nyní si ukážete jiný příklad:

```
9. from microbit import *
10. import random
11. while True:
12.     x = random.randint(0, 4)
13.     y = random.randint(0, 4)
14.     if (display.get_pixel(x, y)):
15.         display.set_pixel(x, y, 0)
16.     else:
17.         display.set_pixel(x, y, 9)
18.     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 0, podmínka není splněna. Následně se nastaví opačná intenzita.

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

PRŮVODCE TEORIÍ

První program – Hello world

Otevřete si editor *mu* a stiskněte tlačítko New. Měli byste vidět následující text:

```
1. from microbit import *
2.
3. # Write your code here :-)
```

Pozor čísla řádků nejsou součástí kódu, ale odpovídají číslům řádku v programu *mu*. V této publikaci uvádím čísla řádků pro snazší odkazování na konkrétní místa ve zdrojovém kódu.

Program by měl končit odřádkováním a na posledním řádku nesmí být žádné znaky (ani mezery).

Na řádku 1 se zavádí systémová knihovna, která zavádí potřebné funkce a metody pro práci s micro:bitem. Tímto řádkem **musí** začínat všechny vaše programy.

Znak # na začátku třetího řádku znamená, že se jedná o komentář. Tento řádek můžete klidně smazat a kód programu psát místo něj. Je možné i psát na řádek 2 anebo pod komentář na řádek 4.

Zkuste pro začátek následující kód:

```
1. from microbit import *
2.
3. display.scroll("Ahoj svete")
```

Popis: řádek 3 znamená, že po displeji micro:bitu má běžet kód uvedený v uvozovkách. Stiskněte tlačítko Save a kód uložte. Programu můžete dát příponu .py (např. ahoj_svete.py). Pokud jí neuvedete, bude mu přiřazena automaticky.

Nyní připojte svůj micro:bit pomocí USB kabelu k počítači. Počkejte asi pět vteřin a pak stiskněte tlačítko Flash. Vyčkejte až přestane blikat žlutá LED dioda na micro:bitu a pak byste měli vidět, jak váš text přeběhne přes displej micro:bitu.

Nyní můžete vyzkoušet následující modifikaci kódu:

```
1. from microbit import *
2.
3. while True:
4.     display.scroll("Ahoj svete")
5.     sleep(1000)
```

Popis: Na řádku 3 je nyní zaveden tzv. nekonečný cyklus. Jeho příkazy jsou odsazené o čtyři mezeričky od začátku řádků. Pozor – je třeba dodržet stejný počet mezer (může být i vyšší, ale násobek čtyř) a nelze použít tabulátor. Na řádku 5 je pak příkaz sleep – čekej 1000 milisekund – 1 sekundu.

Program v nekonečné smyčce vypisuje text a pak čeká jednu sekundu.

Další příklady

Zadání: Napište program, který vypíše čísla od jedné do deseti a pak skončí.

Řešení:

```
1. from microbit import *
2.
3. for i in range(1, 11):
4.     display.scroll(str(i))
```

Popis: Na řádce 3 je zaveden cyklus s pevným počtem opakování. Hodnota proměnné `i` se mění dle rozsahu intervalu `range(a, b)` od `a` do `b-1`. Chcete-li tedy od 1 do 10 musíme psát takto. Za čárkou v intervalu musí být mezera. Pozor na konci řádku je dvojtečka, tady se také často dělá chyba. Na řádce čtyři je pak výpis čísla. Protože funkce `display.scroll` pracuje pouze s řetězci musíte příkazem `str(i)` převést číslo na řetězec.

Zadání: Řešte předchozí příklad pomocí funkce `while`

Řešení:

```
1. from microbit import *
2.
3. i = 1
4. while (i < 11):
5.     display.scroll(str(i))
6.     i = i + 1
```

Popis: Na řádce 3 do proměnné `i` přiřadíte hodnotu 1. Pozor okolo = jsou vyžadovány mezery. Na řádce 4 je cyklus, který se opakuje dokud je `i` menší než 11. Pozor kolem nerovnosti musí být mezery a na konci řádku je dvojtečka. Na řádce 6 zvyšujeme hodnotu proměnné `i` o jedničku. Pozor opět na chybějící mezery.

Zadání: Po dobu jedné vteřiny zobraz na displeji písmeno X.

Řešení:

```
1. from microbit import *
2.
3. display.show("X")
4. sleep(1000)
5. display.clear()
```

Popis: Na řádce 3 zobrazíte písmeno X (opět nutno jako řetězec). Na řádce 4 čeká program jednu sekundu a příkaz na řádce 5 smaže displej.

Přednastavené obrázky

MicroPython obsahuje asi padesát připravených obrázků. Ukázka jejich použití je v následujícím kódu:

```
1. from microbit import *
2.
3. display.show(Image.SAD)
4. sleep(1000)
5. display.show(Image.SMILE)
6. sleep(1000)
7. display.show(Image.HAPPY)
8. sleep(1000)
9. display.clear()
```

Zobrazení obrázků je na řádcích 3, 5 a 7. Jak je vidět, jedná se o konstanty začínající slovem Image.

Seznam všech obrázků naleznete v příloze A anebo v dokumentaci MicroPythonu pro micro:bit.

Příklad: Pomocí konstant obrázků `Image.HEART` a `Image.HEART_SMALL`, simulujte úder srdce.

Řešení:

```
1. from microbit import *
2.
3. for i in range(1, 100):
4.     display.show(Image.HEART)
5.     sleep(400)
6.     display.show(Image.HEART_SMALL)
7.     sleep(400)
8. display.clear()
```


Vlastní obrázky

Příklad: Zobrazte na displeji obrázek rakety

Řešení:

```
9. from microbit import *
10.
11. raketa = Image("00900:"
12.                "05550:"
13.                "05550:"
14.                "09990:"
15.                "90909:")
16.
17. display.show(raketa)
```

Popis: Struktura na řádcích 3 až 7 popisuje obrázek. Pětice čísel ukončených dvojtečkou uzavřená do apostrofů popisuje vždy jeden řádek displeje shora dolů. Číslo pak znamená intenzitu světla od 0 (dioda nesvítí) po 9 (dioda svítí naplno). Na řádku 9 je pak obrázek zobrazen.

Nyní si na základě tohoto příkladu sestojíme pohyblivý obrázek startující rakety. Zdrojový kód je následující:

```
1. from microbit import *
2.
3. raketa1 = Image("00900:"
4.                "05550:"
5.                "05550:"
6.                "09990:"
7.                "90909:")
8.
9. raketa2 = Image("00900:"
10.                "05550:"
11.                "05550:"
12.                "09990:"
13.                "99999:")
14.
15. raketa3 = Image("05550:"
16.                "05550:"
17.                "09990:"
18.                "99999:"
19.                "00000:")
20.
21. raketa4 = Image("09990:"
22.                "99999:"
23.                "00000:"
24.                "00000:"
25.                "00000:")
26.
27. raketa5 = Image("99999:"
28.                "00000:"
29.                "00000:"
30.                "00000:"
31.                "00000:")
32.
```

```

33.raketa6 = Image("00000:"
34.           "00000:"
35.           "00000:"
36.           "00000:"
37.           "00000:")
38.
39.raketa = [raketa1, raketa2, raketa3, raketa4, raketa5,
            raketa6]
40.display.show(raketa, delay=500)

```

Na řádcích 3 až 37 je postupně šest obrázků, označených raketa1 až raketa6. Na řádce 39 je z těchto obrázků sestavena struktura zvaná list (seznam). Ta je pak na řádce 40 postupně zobrazována, kdy mezi jednotlivými snímky je pauza půl sekundy.

Práce s konkrétní diodou

Příklad: Sestrojte program, který bude náhodně rozsvěcet jednotlivé diody s různou intenzitou světla.

Řešení:

```

1.from microbit import *
2.import random
3.
4.while True:
5.    x = random.randint(0, 4)
6.    y = random.randint(0, 4)
7.    intenzita = random.randint(0, 9)
8.    display.set_pixel(x, y, intenzita)
9.    sleep(10)

```

Program používá generátor náhodných čísel. Pro jeho použití je nutné načíst knihovnu random na řádce 2. Na řádcích 5 až 7 je pak tento generátor volán funkcí random.randint, která má dva parametry a, b a vrací náhodné celé číslo z uzavřeného intervalu <a,b>.

Funkce na řádce 8 display.set_pixel má tři parametry x, y, intenzita a nastavuje na souřadnicích x (sloupec) a y (řádek) diodu na intenzitu (0 až 9). Bod 0,0 je vlevo nahoře, vpravo dole pak 4,4.

Použití funkce sleep je nutné jinak dochází k příliš rychlému „blikání“.

Příklad: Upravte předchozí zadání tak, že budete nastavovat pouze dvě úrovně intenzity (0 a 9) a to tak, že budete náhodně vybírat souřadnice a pokud dioda na dané souřadnici nebude svítit, tak jí rozsvítíte a naopak.

Řešení:

```
1. from microbit import *
2. import random
3.
4. while True:
5.     x = random.randint(0, 4)
6.     y = random.randint(0, 4)
7.     if (display.get_pixel(x, y)):
8.         display.set_pixel(x, y, 0)
9.     else:
10.        display.set_pixel(x, y, 9)
11.    sleep(10)
```

Popis: Na řádce sedm je použita funkce `display.get_pixel(x, y)`. Ta vrací hodnotu svícení dané diody. V našem příkladě využíváme toho, že pokud dioda svítí, vrátí hodnotu větší než nula a tudíž je podmínka splněna.

Pozor druhá úroveň odsazení musí být opět násobek čtyř a je tedy osm mezer.

PŘÍLOHA – SEZNAM PŘIPRAVENÝCH OBRÁZKŮ

- `Image.HEART`
- `Image.HEART_SMALL`
- `Image.HAPPY`
- `Image.SMILE`
- `Image.SAD`
- `Image.CONFUSED`
- `Image.ANGRY`
- `Image.ASLEEP`
- `Image.SURPRISED`
- `Image.SILLY`
- `Image.FABULOUS`
- `Image.MEH`
- `Image.YES`
- `Image.NO`
- `Image.CLOCK12`, `Image.CLOCK11`, `Image.CLOCK10`, `Image.CLOCK9`, `Image.CLOCK8`, `Image.CLOCK7`, `Image.CLOCK6`, `Image.CLOCK5`, `Image.CLOCK4`, `Image.CLOCK3`, `Image.CLOCK2`, `Image.CLOCK1`
- `Image.ARROW_N`, `Image.ARROW_NE`, `Image.ARROW_E`, `Image.ARROW_SE`, `Image.ARROW_S`, `Image.ARROW_SW`, `Image.ARROW_W`, `Image.ARROW_NW`
- `Image.TRIANGLE`
- `Image.TRIANGLE_LEFT`
- `Image.CHESSBOARD`
- `Image.DIAMOND`
- `Image.DIAMOND_SMALL`
- `Image.SQUARE`
- `Image.SQUARE_SMALL`
- `Image.RABBIT`
- `Image.COW`
- `Image.MUSIC_CROTCHET`

- `Image.MUSIC_QUAVER`
- `Image.MUSIC_QUAVERS`
- `Image.PITCHFORK`
- `Image.XMAS`
- `Image.PACMAN`
- `Image.TARGET`
- `Image.TSHIRT`
- `Image.ROLLERSKATE`
- `Image.DUCK`
- `Image.HOUSE`
- `Image.TORTOISE`
- `Image.BUTTERFLY`
- `Image.STICKFIGURE`
- `Image.GHOST`
- `Image.SWORD`
- `Image.GIRAFFE`
- `Image.SKULL`
- `Image.UMBRELLA`
- `Image.SNAKE`