

4 POLOHA

Tato kapitola se věnuje orientaci micro:bitu a jeho pohybu v prostoru

Co se naučíte

- Pracovat s vestavěným akcelerometrem, pochopit jeho možnosti a využít jej
- Používat gesta
- Pracovat s vestavěným kompasem
- Používat detekci magnetického pole

Co budete potřebovat

- PC s nainstalovaným editorem Mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukter nebo sluchátka s jackem, popřípadě piezzo buzzer

Časová náročnost

Čtyři vyučovací hodiny po 45 minutách

Akcelerometr

Akcelerometr (Accelerometer) je zařízení pro zjištění aktuální orientace, směru pohybu, zrychlení, popř. volného pádu nebo třesení.

Micro:bit obsahuje přímo integrovaný akcelerometr. Jak je patrné z uvedené definice můžete v našem programu použít akcelerometr pro zjištění polohy micro:bitu nebo typu aktuálního pohybu. S velkým úspěchem jej lze využít v případě, že micro:bit použijete jako ovladač pro jiné zařízení (např. vozítko) řízené jiným micro:bitem. Jak spolu mohou komunikovat vám prozradí následující kapitola.

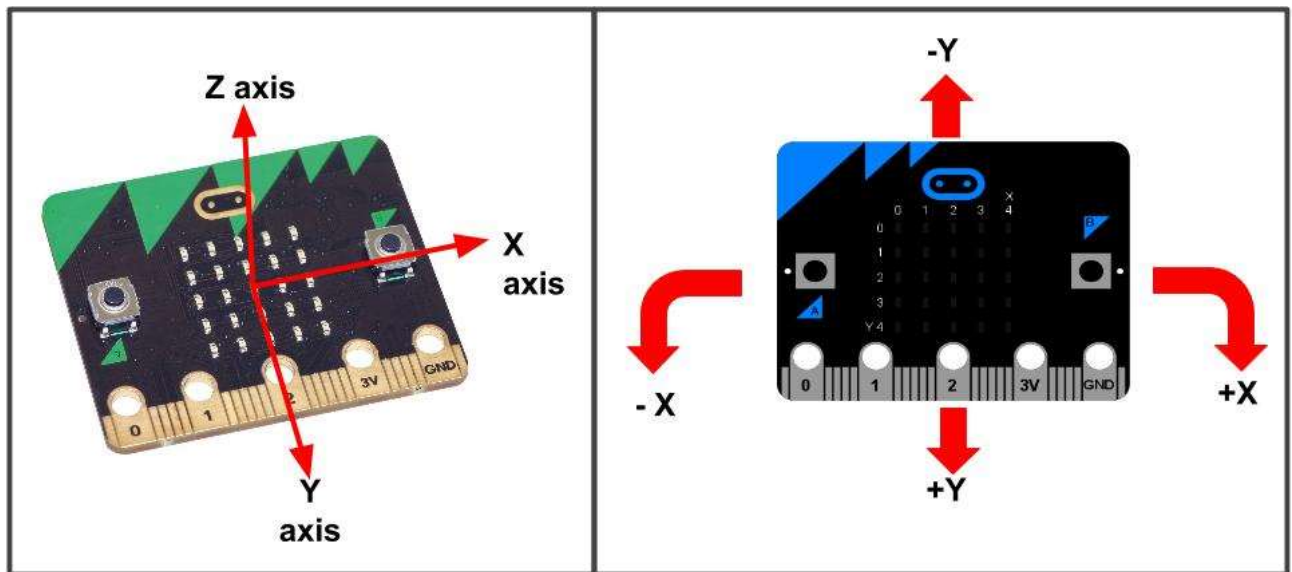
Nejprve se naučíte, jak akcelerometr využít pro určení orientace micro:bitu v prostoru. Pro zjištění této pozice slouží trojice příkazů:

```
accelerometer.get_x(), accelerometer.get_y() a accelerometer.get_z()
```

Tyto tři příkazy vrací hodnoty od -2000 do + 2000.

Pokud micro:bit položíte na stůl diodami nahoru konektory k sobě, pak směr doprava doleva je osa x a směr od sebe k sobě osa y. Náklon vpravo je dán kladnými hodnotami u `accelerometer.get_x()` a vlevo pak zápornými. Náklon vpřed nám dá kladné znaménko u `accelerometer.get_y()` a náklon vzad záporné.

Pokud micro:bit postavíte na konektory diodami k sobě, pak náklon k sobě vám dá kladné hodnoty u `accelerometer.get_z()` a náklon vzad záporné. Natočení vpravo a vlevo po hraně pak obdobně u `accelerometer.get_x()`.



Poloha os a význam plus a minus ve směru os.

Zdroj: <http://fibasile.fabcloud.io/microbit-notebook/accelerometer/>

Vše si můžete dobře vyzkoušet na následujícím příkladu:

```
from microbit import *
mez = 400
while True:
    naklon = accelerometer.get_x()
    if naklon > mez:
        display.show("P")
    elif naklon < -mez:
        display.show("L")
    else:
        display.show("-")
```

Tento program po nahrání do micro:bitu zobrazuje pozici micro:bitu vůči ose x. Je-li micro:bit vodorovně, zobrazuje na displeji znak -, pokud micro:bit nakloním doprava zobrazí P, pokud doleva zobrazí L. Proměnná `mez`, určuje, jak velký sklon bereme jako naklonění vpravo či vlevo.

Nejprve zkuste experimentovat s hodnotou proměnné `mez` a zkoumejte, jak je nutné naklonit micro:bit pro projevení změny.

Nyní změňte hodnotu `x` postupně na `y` a `z` a prohlédněte si, jak reaguje hodnota proměnných na jednotlivé polohy micro:bitu. Protože zde nemá smysl pravá a levá strana, nahraďte např. hodnoty (P, -, L) za (+, 0, -). Ideálně máte-li k dispozici tři micro:bity zkuste do každého nahrát program pro jednu osu a zkoumejte, jak se mění hodnoty. Jak sami vidíte lze takto velice dobře sestavit bezdrátový ovladač, reagující na změnu orientace v prostoru.

V následujícím příkladu si představíme jednoduchý simulátor nástroje *theremin*. Jedná se o nástroj, na který se hraje, aniž by se ho hráč dotýkal, kdy jednou rukou určuje výšku tónu a druhou dobu trvání. Takovýto nástroj postavit nedokážeme, ale budeme jej simulovat tak, že náklonem micro:bitu ve směru osy x budeme určovat výšku tónu a náklonem ve směru osy y délkou tónu.

Pro jednoduchost se spokojíme s tóny v rozsahu jedné oktávy od C4 do C5 a čtyřmi různými délkami 1, 2, 4 a 8 (viz předchozí kapitola). K micro:bitu V1 si připojte repráčky nebo sluchátka na piny 0 a GND a nahrajte následující program:

```
from microbit import *
import music
while True:
    x = accelerometer.get_x()
    y = accelerometer.get_y()
    if (x < -1000):
        ton = "C4"
    elif (x < -700):
        ton = "D4"
    elif (x < -400):
        ton = "E4"
    elif (x < -100):
        ton = "F4"
    elif (x < 200):
        ton = "G4"
    elif (x < 500):
        ton = "A4"
    elif (x < 800):
        ton = "B4"
    else:
        ton = "C5"
    if (y < -500):
        nota = ton
    elif (y < 0):
        nota = ton + ":2"
    elif (y < 500):
        nota = ton + ":4"
    else:
        nota = ton + ":8"
    music.play(nota)
```

Všimněte si, jak postupně vzniká řetězec, tvořený výškou a délkou noty, který je vzápětí přehrán. Experimentujte se zvětšením (zmenšením) rozsahu výšek a délek tónů.

Gesta

Gesta (gesture) u micro:bitu jsou nějaké pohybové činnosti, které se s ním v daném okamžiku dějí. Může se jednat o otočení požadovaným směrem, zrychlení, zatřesení a volný pád. Seznam gest naleznete v příloze k této kapitole.

Začneme jednoduchým programem, který zobrazí na displeji úsměv, pokud je micro:bit otočen displejem vzhůru (face_up) a naopak smutný obličej, je-li tomu jinak:

```

from microbit import *
while True:
    gesture = accelerometer.current_gesture()
    if gesture == "face up":
        display.show(Image.HAPPY)
    else:
        display.show(Image.ANGRY)

```

Můžete zkusit i jiné gesto z dostupných. Pouze pozor na volný pád, aby nedošlo k poškození micro:bitu. Abyste stihli gesto, můžete obrázek nechat na displeji nějakou dobu např. 3 sekundy.

Rovněž lze použít následující metody. GESTO nahrad'te libovolným gestem dle přílohy.

`microbit.accelerometer.current_gesture()` – vrací jméno právě použitého gesta

`microbit.accelerometer.is_gesture(GESTO)` – vrací True nebo False podle toho, zda právě probíhá GESTO

`microbit.accelerometer.was_gesture(GESTO)` – vrací True nebo False podle toho, zda od posledního dotazu bylo GESTO

`microbit.accelerometer.get_gestures()` – vrací seznam gest od posledního zjišťování gesta nebo od spuštění programu

Následující příklad vám poskytne všeteckou odpověď na problém, který vás trápí. Myslete usilovně na problém, s jehož řešením si nevíte rady, pak zatřeste micro:bitem a on vám poradí.

```

from microbit import *
import random
odpovedi = [
    "Jiste",
    "Urcite",
    "Pravdepodobne",
    "To vypada dobre",
    "Ano",
    "Zeptej se pozdeji",
    "Ted nevim",
    "Nelze urcit",
    "Urcite ne",
    "Ne",
    "Nikdy",
]
while True:
    display.show('8')
    if accelerometer.was_gesture("shake"):
        display.clear()
        sleep(1000)
        display.scroll(random.choice(odpovedi))
        sleep(10)

```

Samozřejmě si upravte odpovědi dle sebe. Jedná se vlastně o úpravu hry magic 8ball. Proto v klidovém stavu zobrazuje micro:bit číslo 8.

Kompas

Micro:bit obsahuje integrovaný kompas, který současně lze použít jako čidlo intenzity magnetického pole.

Základní použití si můžete ukázat na následujícím programu:

```
from microbit import *
compass.calibrate()
while True:
    display.scroll(compass.heading())
    sleep(1000)
```

Před použitím jakýchkoliv funkcí týkajících se magnetického pole je pro správnou funkci nutná kalibrace micro:bitu.

Pro kalibraci je nutno otáčet micro:bitem tak dlouho, než displej zaplníme svítícími diodami. Na Micro:bitu vám vždy před kalibrací proběhne instrukce, jak postupovat. Po zaplnění displeje je třeba několik vteřin (cca. 5) počkat, než se na displeji objeví smajlík.

Pokud přesto kompas micro:bitu ukazuje nečekanou hodnotu, stiskněte na něm klávesu reset a opakujte kalibraci. To obvykle pomůže.

Micro:bit položte na rovnou plochu nebo jej držte co nejvíce rovně. Micro:bit nyní ukáže na displeji azimut (úhel svírající se směrem na sever ve směru hodinových ručiček). Směr azimutu je přímo od displeje nahoru.

V astronomii se úhel určuje od jihu ve směru hodinových ručiček. Upravte program tak aby ukazoval astronomický azimut.

Vyzkoušejte si rovněž, co se stane, když kolem micro:bitu pohybujeme magnetem nebo zmagnetizovaným předmětem (nůžky, šroubovák ...).

Nyní program upravíme tak, aby micro:bit ukazoval symboly světových stran S, V, J, Z. Za sever budeme považovat intervaly úhlů (0,45) a (316, 359), za východ (46, 135), za jih (136, 225) a za západ (226, 315).

```
from microbit import *
compass.calibrate()
while True:
    uhel = compass.heading()
    if (uhel < 46):
        display.show("S")
    elif (uhel < 136):
        display.show("V")
    elif (uhel < 226):
        display.show("J")
    elif (uhel < 316):
        display.show("Z")
    else:
        display.show("S")
    sleep(1000)
```

Program nyní upravíme tak, aby ukazoval na displeji micro:bitu směr na sever. Využijeme při tom obrázek `Image.ALL_CLOCKS`. Jedná se vlastně o pole dvanácti obrázků, které se volají `Image.ALL_CLOCKS[uhel]`, kde `uhel` je číslo od 0 do jedenácti. Na displeji pak ukazují čáru (lépe křivku) od středu micro:bitu ve směru malé hodinové ručičky ukazující danou hodinu o hodnotě proměnné `uhel`. Pozor namísto 12 směr nahoru ukazuje hodnota 0. Můžete si to ověřit následujícím programem:

```
from microbit import *
for uhel in range(0, 12):
    display.show(Image.ALL_CLOCKS[uhel])
    sleep(1000)
    display.clear()
```

Nyní zůstává otázkou, jak zajistit, aby na displeji byl zobrazen směr k severu. Máme celkem dvanáct poloh ručičky (ukazatele). To znamená 30° na každou polohu, např. sever je od -15° do 15° . Pozici ručičky pak dostaneme na první pohled krkolomným vzorcem:

`Pozice = ((15 - Azimut) // 30) % 12`

Máme zde pro vás možná neznámé operace `//` a `%`. Jejich význam je následující:

`//` je celočíselné dělení – dělení beze zbytku. Např. $7 // 2 = 3$ a $7 // 3 = 2$.

`%` je zbytek po dělení. Např. $7 \% 2 = 1$, $7 \% 3 = 1$.

Výpočtem `(15 - Azimut) // 30` dostaneme číslo od 0 do -11. (pro 0° dostaneme 0 a pro 359° dostaneme -11). Tyto hodnoty převedeme na kladné pomocí operace `% 12` a dostaneme správný index pro pozici ukazatele.

Vzorec lze samozřejmě upravit. Například místo 15 lze použít hodnotu 375 a místo `% 12` lze použít `+ 12`. V těchto případech, je ale výsledek v intervalu 1 až 12 a hodnotu 12 je třeba převést na 0 (hodnota indexu 12 je mimo rozsah).

Program tak vypadá následovně:

```
from microbit import *
compass.calibrate()
while True:
    uhel = ((compass.heading() - 15) // 30)
    display.show(Image.ALL_CLOCKS[uhel])
```

Intenzita magnetického pole

Na závěr si ukážeme ještě další vlastnost, kterou má kompas. Umožňuje rovněž měřit hodnotu magnetického pole v jednotkách nT (nano tesla).

Můžeme tedy napsat následující program, který sleduje, zda magnetické pole v okolí překročí určitou hodnotu (zde 5000 nT) a pak zobrazit na určitou dobu smajlík.

```
from microbit import *
hodnota = 5000
compass.calibrate()
pocatek = compass.get_field_strength()
while True:
    sleep(100)
    sila = compass.get_field_strength()
    if abs(sila - pocatek) > hodnota:
        display.show(Image.HAPPY)
        sleep(3000)
        display.clear()
```

Vyzkoušejte, v okolí, kterých přístrojů se nachází magnetické pole. Např. počítače, mobily, tablety. Rovněž také zmagnetizované nůžky, nože anebo šroubováky.

S pomocí tohoto programu můžete předvést následující kouzlo. V ruce ukryjete malý silný magnet a přejedete touto rukou nad micro:bitem. Micro:bit zobrazí úsměv. Řekněte neznalému, že micro:bit se rozsvítí pouze v okolí lidí s magnetickým potenciálem a nechte je pohyb zopakovat. Bez magnetu samozřejmě k ničemu nedojde.

PŘÍLOHA – SEZNAM PŘIPRAVENÝCH GEST

- up – Micro:bit je otočen nahoru
- down – Micro:bit je otočen dolů
- left – Micro:bit je otočen vlevo
- right – Micro:bit je otočen vpravo
- face up – Micro:bit leží otočen diodami nahoru
- face down – Micro:bit leží otočen diodami dolů
- freefall – Micro:bit padá volným pádem
- 3g – Micro:bit se pohybuje zrychlením 3g
- 6g – Micro:bit se pohybuje zrychlením 6g
- 8g – Micro:bit se pohybuje zrychlením 8g (pravděpodobně se nalézá ve startující raketě)
- shake – Micro:bitem je třeseno