

4 POLOHA

Co se naučíte

- Tato kapitola se věnuje orientaci a pohybu micro:bitu v prostoru
- 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 micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukční nebo sluchátka s jackem, popřípadě piezzo buzzer.

PRŮVODCE HODINOU IV-1

Studenti se v této hodině naučí jakým způsobem zjistit orientaci micro:bitu v prostoru a jak tuto informaci využít.

Co bude v této hodině potřeba:

- PC s editorem mu.
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Reprodukter nebo sluchátka s jackem, popřípadě piezzo buzzer.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

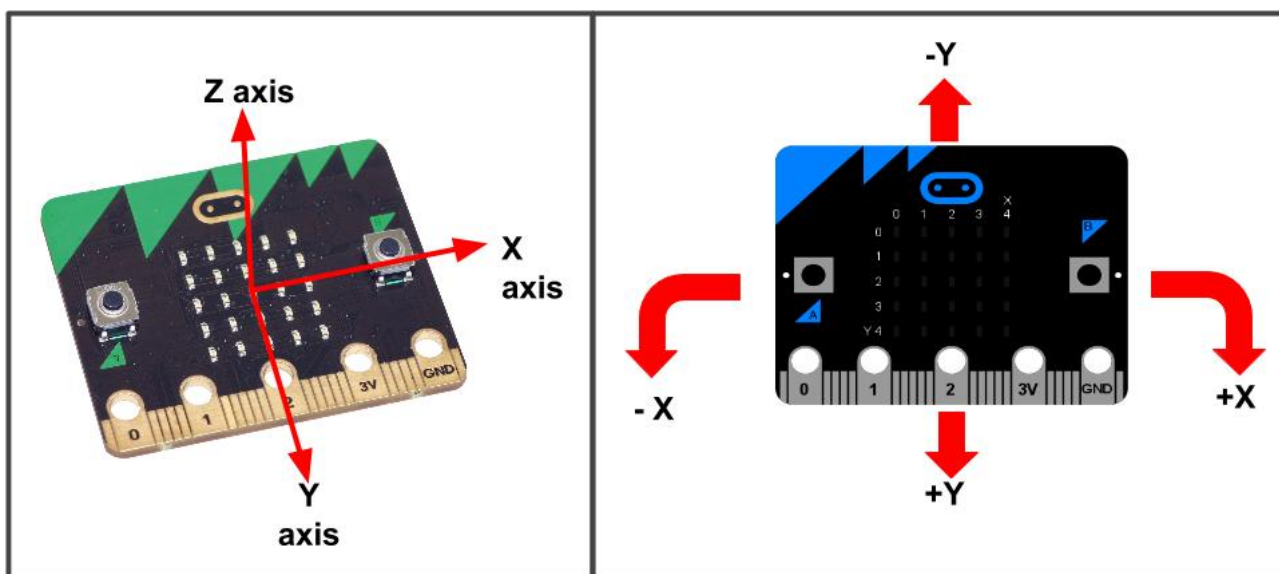
1. krok 25 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať si připraví sluchátka. Raději mějte připravená sluchátka pro ty, kteří si je zapomenou.

Vysvětlete studentům pojem akcelerometr – zařízení pro zjištění aktuální orientace, směru pohybu, zrychlení, popř. volného pádu nebo zrychlení.

Zeptejte se studentů na využití akcelerometru – herní ovladač, generátor událostí. Program reaguje na zatřesení, změnu orientace.

Vysvětlete studentům, jak fungují osy x , y a z u micro:bitu. Je-li položen vodorovně, piny k vám, je osa x rovnoběžně s vámi (a sleduje tedy náklon vlevo a vpravo), osa y pak směrem k vám a sleduje náklon od vás k vám. Osa z v tomto případě není sledovatelná. Použijete jí, pouze pokud je micro:bit „postaven“ na konektorech a naklání se k vám a od vás.



3.

Poloha os a význam plus a minus ve směru os.

Zapište a odlaďte následující program, ukazující pohyb ve směru osy x:

```
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 ukazuje princip sledování pohybu micro:bitu ve směru osy x. Proměnná `mez` určuje hodnotu, od které považujeme micro:bit za nakloněný vpravo či vlevo.

Nechte studenty sledovat, jak program funguje vzhledem k pozici micro:bitu v prostoru. Nechte je měnit hodnotu proměnné `mez`. Poté změňte osy. Zcela vždy postačí nahradit `get_x` za `get_y` nebo `get_z` a naopak. Toto je vlastně způsob, jak můžete micro:bit použít jako dálkový ovladač, až se naučíte v některé z příštích hodin, přenášet data mezi micro:bity.

2. krok 20 minut

Nyní si postavíte jednoduchý simulátor Thereminu. Theremin je nástroj, který ovládáme pohybem rukou bez dotyku nástroje. Pohybem jedné ruky určujeme výšku tónu a pohybem druhé, pak jeho délku.

Zapište a odlaďte 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)
```

Připojte k micro:bitu repráčky (sluchátka), dle postupu z minulých lekcí (mezi piny 0 a GND). Natáčením micro:bitu vpravo a vlevo regulujete výšku tónu, dopředu dozadu jeho délku. Ponechte studenty upravovat rozsah tónů, citlivost atd.

PRACOVNÍ LIST IV-1

Co se naučíte

- Co je to akcelerometr a jak funguje
- Naučit sledovat natočení micro:bitu v prostoru
- Využít akcelerometr jako ovladač micro:bitu

Co budete potřebovat

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

A jděte na to ...

Pokud neznáte pojem akcelerometr, nechte si jej vysvětlit vyučujícím.

K jakým účelům byste použili akcelerometr?

Napište a odlaďte následující program:

```
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 sleduje náklon micro:bitu dle osy *x* (vlevo a vpravo). Proměnná *mez* určuje, jaký náklon budeme považovat za mezní, abychom řekli, že je micro:bit nakloněn vpravo či vlevo. Experimentujte s tím jak se micro:bit chová dle orientace v prostoru při různém natočení. Zkuste nahradit `get_x` za `get_y` popřípadě `get_z` aby jste vyzkoušeli orientaci vůči ose *y* nebo *z*.

Nyní si vyzkoušíte simulaci hudebního nástroje Theremin. Pokud nevíte co je Theremin, zeptejte se vyučujícího nebo si jej najděte na internetu.

Napište 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)
```

Připojte k micro:bitu sluchátka (repráčky) podobně jako v minulé hodině (mezi piny 0 a GND). Měli byste nyní slyšet tón. Otáčením micro:bitu vlevo a vpravo měníte výšku tónu, od sebe k sobě jeho délku.

Experimentujte se změnou rozsahů (zvětšení či zmenšení tónového rozsahu).

Všimněte si v programu, jakým způsobem se v Pythonu spojují dva řetězce. Jedná se vlastně o sčítání.

PRŮVODCE HODINOU IV-2

V této hodině se studenti naučí pracovat s gesty.

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 – 5 minut

Vysvětlíte studentům, co jsou to gesta – nějaké konkrétní pohyby micro:bitu. Otočení, zatřesení, volný pád atd. Jejich úplný seznam včetně jmen je v příloze k této kapitole.

Zeptejte se studentů, jak by jednotlivá gesta mohli využít k různým činnostem.

2. krok – 15 minut

Napište a odlaďte následující program:

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

Tento program zobrazuje smajlík dle orientace micro:bitu v prostoru. Pouze je-li micro:bit obrácen displejem vzhůru a je v klidu zobrazuje se úsměv, jinak se zobrazí zamračená tvář.

Nechte studenty vyzkoušet i jiná gesta a smajlíky. Pokud budou testovat volný pád, měli by si dát pozor na rozbití micro:bitu. Poradte jim, aby v případě zobrazení smajlíku např. při volném pádu přidali pauzu např. tři sekundy, aby byl smajlík chvíli zobrazen.

3. krok – 25 minut

Nechte studenty zapsat a odladit následující program. Jedná se o variaci hry Magic 8 ball. Micro:bit v případě zatřesení na displeji zobrazuje náhodné odpovědi. Řekněte studentům, že se mají v duchu soustředit na nějaký problém, pak požádat micro:bit o jeho řešení a zatřást s ním.

Program je delší – zvažte následující možnosti. Program studentům nahrajte na místo, kde k němu budou mít přístup anebo jej připravte pouze s možnostmi ano, ne, možná jako je to na prezentaci a nechte studenty doplnit další možnosti. Nezapomeňte připomenout, že nelze používat háčky a čárky.

Nechte studenty libovolně upravovat možnosti.

```
from microbit import *
import random
odpovedi = [
    "Jiste",
    "Urcite",
    "Bez obav",
    "Ano, ano, ano",
    "Uvidime, uvidime",
    "Pravdepodobne",
    "To vypada dobre",
    "Ano",
    "Zeptej se pozdeji",
    "Ted nevim",
    "Nelze urcit",
    "Radeji ne",
    "Me vnitřni já říká ne",
    "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)
```


PRACOVNÍ LIST IV-2

Co se naučíte

- Co jsou to gesta
- Jaká gesta umí micro:bit zaznamenat
- Jak na micro:bitu ovládat gesta a jak je použít.

Co budete potřebovat

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

A jděte na to ...

Otázky:

Co jsou to gesta? Jaké znáte?

Prohlédněte si seznam gest. Jak byste mohli jednotlivá gesta využít?

Napište a odlaďte následující program:

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

Tento program zobrazuje smajlík dle orientace micro:bitu v prostoru. Pouze je-li micro:bit obrácen displejem vzhůru a je v klidu zobrazuje se úsměv, jinak se zobrazí zamračená tvář.

Vyzkoušejte i jiná gesta a jiné smajlíky. Pozor na zkoušení volného pádu atd. Nerozbijte si micro:bit. Můžete přidat časovou pauzu např. 3 sekundy po zobrazení gesta, abyste jej stihli.

Zkuste program upravit tak, aby zobrazoval úsměv anebo nic.

Nyní napište (nahrajte) a odlaďte následující program. Jedná se o variaci hry Magic 8 ball. Micro:bit.

Myslete na nějakou otázku, která vás trápí a pak zatřeste micro:bitem a on vám na vaši otázku odpoví. Jak program funguje?

Zkuste přidat další možnosti. Pozor, jak již víte, nesmíte používat háčky a čárky.

```
from microbit import *
import random
odpovedi = [
    "Ano",
    "Ne",
    "Nevim",
]
while True:
    display.show('8')
    if accelerometer.was_gesture('shake'):
        display.clear()
        sleep(1000)
        display.scroll(random.choice(odpovedi))
    sleep(10)
```

PRŮVODCE HODINOU IV-3

V této hodině se studenti naučí pracovat s micro:bitem jako s kompasem a stanovit pomocí něj azimut.

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

Kroky

Látku této hodiny pravděpodobně nestihnete. V takovémto případě klidně zbytek odložte na další hodinu, která je naopak krátká. Z tohoto důvodu není ani tato (a příští) hodina dělena na kroky.

Proberte se studenty pojmy kompas a azimut. Zeptejte se, zda těmto pojmům rozumí.

Micro:bit obsahuje integrovaný kompas, který současně lze použít jako čidlo intenzity magnetického pole. Tento kompas je nutné vždy před použitím kalibrovat, jinak nelze ručit za jeho správnou funkci. Základní použití si můžete ukázat na následujícím programu:

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

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. Micro:bit položte na rovnou plochu nebo jej držte co nejvíce rovně. Micro:bit nyní ukáže na displeji *azimut*. Směr azimutu je přímo od displeje nahoru.

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

Nyní program upravte 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 upravte tak aby ukazoval na displeji micro:bitu směr na sever. Využijte 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 pro hodinu o hodnotě proměnné `uhel`. Pozor namísto 12 směr nahoru ukazuje hodnota 0. Můžete si to ověřit následujícím programkem:

```

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

```

Program, který ukazuje směr sever je v následujícím výpisu:

```

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

```

Pokud věříte tomu, že to vaši studenti pochopí vysvětlíte jim výpočet na čtvrtém řádku. Je vysvětlen v teoretickém základu. Jinak pouze studentům řekněte, že se jedná o výpočet severu z daného azimutu.

Proberte se studenty význam operací `//` (dělení beze zbytku) a `%` (zbytek po dělení).

Otázky (nejen) pro zvědavé:

- Lze nahradit číslo 15, číslem 375? Jak musíme program upravit?
- Můžeme místo `% 12` napsat `+ 12`? Jak musíme program upravit?

Úkoly:

- Upravte program pro zobrazení severu, pomocí šípek `Image.ARROW_N`, `Image.ARROW_NE` atd.
- Upravte program pro zobrazení azimutu, tak aby zobrazoval astronomický azimut 0 je na jihu, 90 západ, 180 sever, 270 východ.

PRACOVNÍ LIST IV-3

Co se naučíte

- Jak pracovat s micro:bitem jako s kompasem
- Co je to azimut a jak jej pomocí micro:bitu stanovit

Co budete potřebovat

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

A jděte na to ...

Vysvětlete pojmy kompas a azimut.

Micro:bit obsahuje integrovaný kompas, který současně lze použít jako čidlo intenzity magnetického pole. Tento kompas je nutné vždy před použitím kalibrovat, jinak nelze ručit za jeho správnou funkci. Základní použití si můžete ukázat na následujícím programu:

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

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. Micro:bit položte na rovnou plochu nebo jej držte co nejvíce rovně. Micro:bit nyní ukáže na displeji *azimut*. Směr azimutu je přímo od displeje nahoru.

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

Nyní program upravte 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 upravte tak aby ukazoval na displeji micro:bitu směr na sever. Využijte 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 pro hodinu o hodnotě proměnné `uhel`. Pozor namísto 12 směr nahoru ukazuje hodnota 0. Můžete si to ověřit následujícím programkem:

```

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

```

Program, který ukazuje směr sever je v následujícím výpisu:

```

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

```

Otázky a úkoly:

- Lze nahradit číslo 15, číslem 375? Jak musíme program upravit?
- Můžeme místo `% 12` napsat `+ 12`? Jak musíme program upravit?
- Upravte program pro zobrazení severu, pomocí šípek `Image.ARROW_N`, `Image.ARROW_NE` atd.
- Upravte program pro zobrazení azimutu, tak aby zobrazoval astronomický azimut 0 je na jihu, 90 západ, 180 sever, 270 východ.

PRŮVODCE HODINOU IV-4

V této hodině se studenti naučí pomocí micro:bitu měřit intenzitu magnetického pole.

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

Postup

Zeptejte se studentů, jaká je jednotka intenzity magnetického pole (magnetické indukce). Je to Tesla. V praxi se používají její dílčí jednotky. Micro:bit umí pomocí kompasu měřit intenzitu magnetického pole v jednotkách nano Tesla (nT)

Můžeme tedy napsat následující program, který změří hodnotu magnetického pole, a pak zjišťuje zda absolutní hodnota změny magnetického pole v okolí překročí určitou hodnotu (zde 5000 nT). Pokud ano, tak zobrazí 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 přístroj 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 rozvíjí pouze v okolí lidí s magnetickým potenciálem a nechte je pohyb zopakovat. Bez magnetu samozřejmě k ničemu nedojte.

PRACOVNÍ LIST IV-4

Co se naučíte

- Pomocí micro:bitu měřit intenzitu magnetického pole.

Co budete potřebovat

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

A jděte na to ...

Otázka: Co je to za jednotku nT (nano Tesla) a pro co se používá?

Můžeme tedy napsat následující program, který změří hodnotu magnetického pole, a pak zjišťuje, zda absolutní hodnota změny magnetického pole v okolí překročí určitou hodnotu (zde 5000 nT). Pokud ano, tak zobrazí 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 přístroj 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 rozvíjí pouze v okolí lidí s magnetickým potenciálem a nechte je pohyb zopakovat. Bez magnetu samozřejmě k ničemu nedojte.

PRŮVODCE TEORIÍ

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 z definice patrné můžete v našem programu použít akcelerometr pro zjištění polohy micro:bitu nebo typu aktuálního pohybu. S velkým úspěchem toho lze využít v případě, že micro:bit použijete jako ovladač pro jiný micro:bit, který např. řídí vozítko.

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

```
accelerometer.get_x()
```

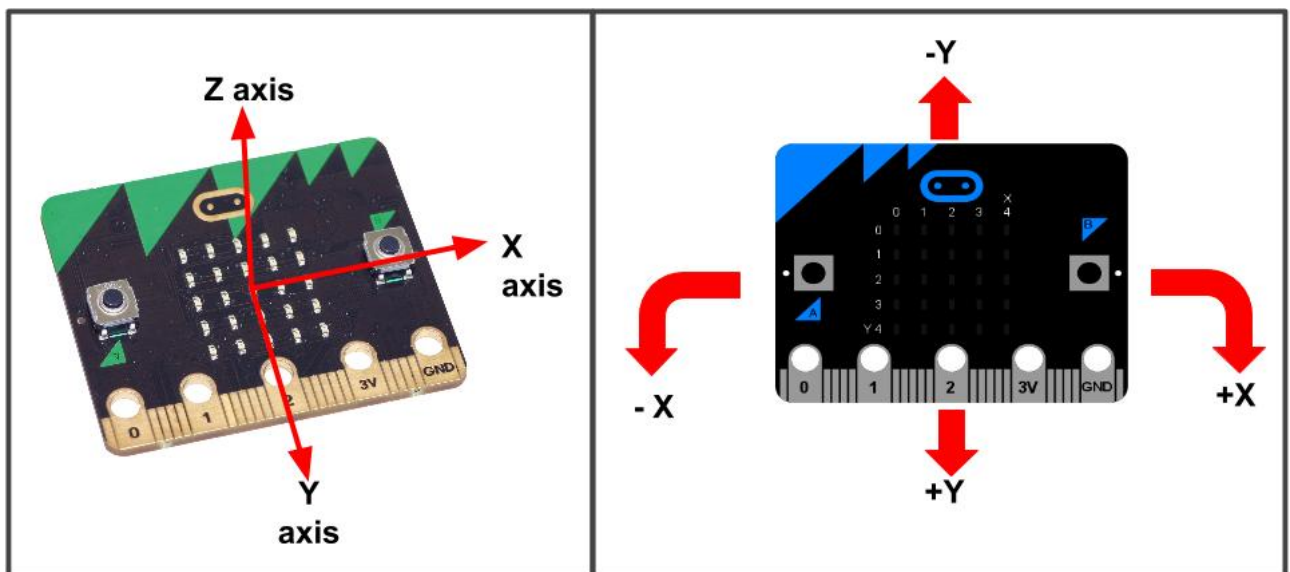
```
accelerometer.get_y()
```

```
accelerometer.get_z()
```

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

Pokud micro:bit položíte na stůl diodami nahoru a 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()`.



4.

Poloha os a význam plus a minus ve směru os.

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 -, 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 sestrojit 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 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 nahraďte 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 poslední gesto bylo GESTO

`microbit.accelerometer.get_gestures()` – vrací seznam gest od posledního

Následující příklad vám poskytne věšteckou 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",
    "Bez obav",
    "Ano, ano, ano",
    "Uvidime, uvidime",
    "Pravdepodobne",
    "To vypada dobre",
    "Ano",
    "Zeptej se pozdeji",
    "Ted nevim",
    "Nelze urcit",
    "Radeji ne",
    "Me vnitřní já říká ne",
    "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 8-ball. Proto v klidovém stavu zobrazuje micro:bit číslo 8.

Kompas

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

Tento kompas je nutné vždy před použitím kalibrovat, jinak nelze ručit za jeho správnou funkci.

Základní použití si můžeme ukázat na následujícím programu:

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

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. Tato kalibrace včetně úvodních instrukcí musí proběhnout vždy před použitím kompasu, jinak nelze zajistit, že bude kompas správně fungovat.

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 programkem:

```
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 $(15 - \text{Azimut}) // 30$ 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 hodnotě 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 přístroj 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 rozvítí pouze v okolí lidí s magnetickým potenciálem a nechte je pohyb zopakovat. Bez magnetu samozřejmě k ničemu nedojte.

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