Vše si můžete dobře vyzkoušet na následujícím příkladu:

```
1. from microbit import *
2. mez = 400
3. while True:
4.     naklon = accelerometer.get_x()
5.     if naklon > mez:
6.         display.show("P")
7.     elif naklon < -mez:
8.         display.show("L")
9.     else:
10. 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 délku 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:

```
1. from microbit import *
2. import music
3. while True:
4.     x = accelerometer.get_x()
5.     y = accelerometer.get_y()
6.     if (x < -1000):
7.         ton = "C4"
8.     elif (x < -700):
9.         ton = "D4"
10.     elif (x < - 400):
11.         ton = "E4"
12.     elif (x < -100):
13.         ton = "F4"
14.     elif (x < 200):
15.         ton = "G4"
16.     elif (x < 500):
17.         ton = "A4"
18.     elif (x < 800):
19.         ton = "B4"
```

```
20.     else:
21.         ton = "C5"
22.     if (y < -500):
23.         nota = ton
24.     elif (y < 0):
25.         nota = ton + ":2"
26.     elif (y < 500):
27.         nota = ton + ":4"
28.     else:
29.         nota = ton + ":8"
30.     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:

```
1. from microbit import *
2. while True:
3.     gesture = accelerometer.current_gesture()
4.     if gesture == "face up":
5.         display.show(Image.HAPPY)
6.     else:
7.         display.show(Image.ANGRY)
```

Můžete zkusit i jiné gesto z dostupných. Pouze pozor na volný pád, aby nedošlo k poškození micro:bitu. Abyste sttihli 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.

```
1. from microbit import *
2. import random
3. odpovedi = [
4.     "Jiste",
5.     "Urcite",
6.     "Bez obav",
7.     "Ano, ano, ano",
8.     "Uvidime, uvidime",
9.     "Pravdepodobne",
10.       "To vypada dobre",
11.       "Ano",
12.       "Zeptej se pozdeji",
13.       "Ted nevim",
14.       "Nelze urcit",
15.       "Radeji ne",
16.       "Me vnitrni ja rika ne",
17.       "Urcite ne",
18.       "Ne",
19.       "Nikdy",
20.   ]
21.   while True:
22.       display.show('8')
23.       if accelerometer.was_gesture('shake'):
24.           display.clear()
25.           sleep(1000)
26.           display.scroll(random.choice(odpovedi))
27.   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:

```
1. from microbit import *
2. compass.calibrate()
3. while True:
4.         display.scroll(compass.heading())
5.         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).

```
1. from microbit import *
2. compass.calibrate()
3. while True:
4.     uhel = compass.heading()
5.     if (uhel < 46):
6.         display.show("S")
7.     elif (uhel < 136):
8.         display.show("V")
9.     elif (uhel < 226):
10.            display.show("J")
11.       elif (uhel < 316):
12.           display.show("Z")
13.       else:
14.           display.show("S")
15.  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 prográmkem:

```
1. from microbit import *
2. for uhel in range(0, 12):
3.     display.show(Image.ALL_CLOCKS[uhel])
4.     sleep(1000)
5. 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 hodnotě intervalu 1 až 12 a hodnotu 12 je třeba převést na 0 (hodnota indexu 12 je mimo rozsah).

Program tak vypadá následovně:

```
1. from microbit import *
2. compass.calibrate()
3. while True:
4.     uhel = ((compass.heading()-15) // 30)
5. 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.

```
1. from microbit import *
2. hodnota = 5000
3. compass.calibrate()
4. pocatek = compass.get_field_strength()
5. while True:
6.     sleep(100)
7.     sila = compass.get_field_strength()
8.     if abs(sila - pocatek) > hodnota:
9.         display.show(Image.HAPPY)
10.          sleep(3000)
11. 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 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