

Využitie senzora teploty a intenzity svetla pre mikrokontrolér micro:bit

Mgr. Martin Cápaj, PhD.



**komenského
inštitút**

BBC micro:bit má zabudovaný senzor teploty, ktorý detekuje aktuálnu teplotu zariadenia v stupňoch celzia. Hodnota teploty je dostupná v jazyku Python ako aj v MakeCode.

Micro:bit v skutočnosti nemá dedikovaný teplotný senzor. V skutočnosti ide len o teplotu silikónovej matrice na hlavnom procesore. ARM jadro procesora všeobecne „beží“ za studena a tak je jeho teplota vhodnou aproximáciou teploty okolia.

Keďže teplotu nie je možné nastaviť ale len čítať, tak ju nájdeme v kategórii vstupov (Input). Príkaz `temperature` len poskytne dáta aproximované z teploty procesora. Je možné ho vV ypísať, či použiť v logickom výraze alebo priradiť jeho hodnoty do premennej.

Úloha 1 Zobraz hodnotu teploty na displej microbita.

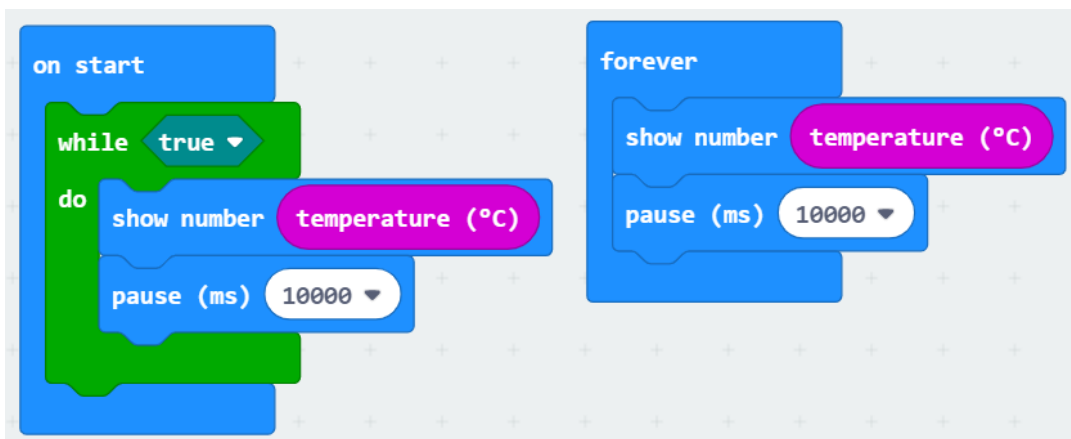


```
from microbit import *  
display.scroll(str(temperature()))
```

MakeCode robí konverziu hodnôt automaticky. V Pythone na konverziu musí myslieť programátor.

Úloha 2 Každých 10 sekúnd zobraz teplotu na displej microbita.

V predchádzajúcej úlohe došlo k zobrazeniu aktuálnej teploty len jednorazovo, a to pri spustení programu v micro:bit. Ak chceme proces opakovať je potrebné zapojiť cyklus (slučku). Ako najlepšie riešenie je opakovať vypisovanie donekonečna.



V MakeCode máme dve možnosti. Buď po štarte spustíme nekonečnú slučku, cyklus `while true do`, alebo vyberieme udalosť `Forever`. Hlavná časť programu je pre obe riešenie rovnaká. Avšak využitie cyklus `while` viac pripomína riešenie, ktoré zrealizujeme v jazyku Python.

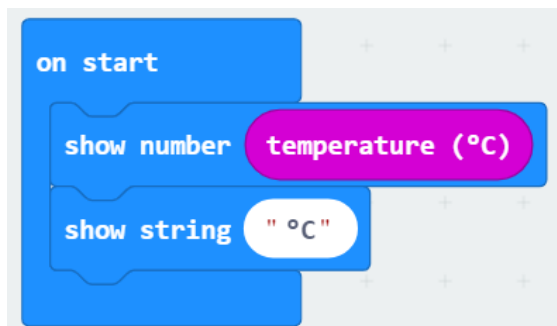
```
from microbit import *
```

```
while True:
    display.scroll(str(temperature()))
    sleep(10000)
```

Úloha 3 Zobraz hodnotu teploty na displej microbita a pridaj aj informáciu v akých jednotkách meriame, použijeme stupne Celsia.

Stupnica Celsia, známa tiež ako stupnica centigrade, je teplotná stupnica používaná Medzinárodným systémom jednotiek (SI). Ako jednotku odvodenú od SI používajú všetky krajiny okrem Spojených štátov, Bahám, Belize, Kajmanských ostrovov a Libérie. Je pomenovaný podľa švédskeho astronóma Andersa Celsia (1701 - 1744), ktorý vyvinul podobnú teplotnú stupnicu. Stupeň Celzia (symbol: ° C) sa vzťahuje na špecifickú teplotu na stupnici stupňov Celzia. Pred premenovaním na počesť Andersa Celsia v roku 1948 bola jednotka nazývaná centigrade, z latinského centa, čo znamená 100 a gradus, čo znamená kroky. Od roku 1743 je stupnica Celsia založená na 0 ° C pre bod mrazu vody a 100 ° C pre teplotu varu vody. Pred rokom 1743 bola stupnica tiež založená na bodoch varu a teploty topenia vody, ale hodnoty boli obrátené (t. J. Teplota varu bola 0 ° C a teplota topenia bola 100 ° C). Obrátenie stupnice v roku 1743 navrhol Jean-Pierre Christin.

Potrebuje vypísať dve informácie rôzneho dátového typu. Prvá je číselná hodnota (teplota) a druhá je reťazec (°C). V prostredí MakeCode preto musíme použiť dva príkazy.



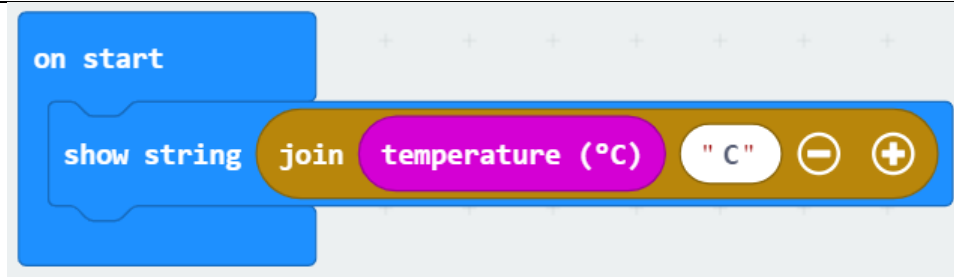
```
from microbit import *

display.scroll(str(temperature()))
display.scroll("°C")
```

Zobrazovanie hodnoty "°C" nebude fungovať. Nepodporovaný znak „stupeň“ sa nahradí prázdny znakom.

Riešenie však bude vykazovať neželané správanie. Najskôr sa totiž zobrazí hodnota a až potom keď hodnota „zmizne“ z displeja sa zobrazí v akých jednotkách meriame. Potrebovali by sme však, aby sa obe hodnoty zobrazovali bezprostredne po sebe, preto je nutné ich spojiť do jedného výpisu.

V prostredí MakeCode musíme použiť príkaz `join` z ponuky `Text`. V prípade Pythonu stačí použiť príkaz zreťazenia.

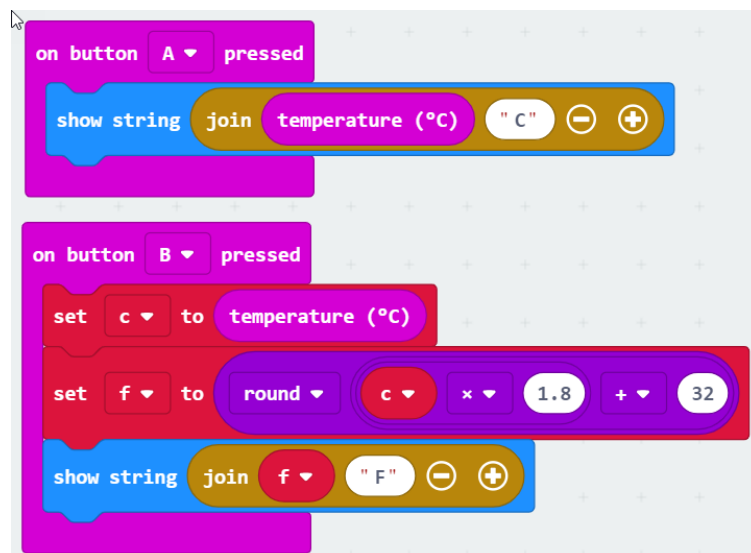


```
from microbit import *  
display.scroll(str(temperature() + "C ")
```

Úloha 4 Po stlačení tlačidla A sa na displeji zobrazí hodnota teploty v stupňoch Celzia, po stlačení tlačidla B sa na displeji zobrazí hodnota teploty v stupňoch Fahrenheit.

Stupnica Fahrenheita je teplotná škála založená na mierke navrhutej v roku 1724 holandsko-nemecko-poľským fyzikom menom Daniel Gabriel Fahrenheit (1686–1736). Používa jednotku Fahrenheita (symbol: ° F). Dolným bodom definovania, 0 ° F, je teplota zmrazenia roztoku soľanky vyrobenej z rovnakých častí ľadu, vody a soli (chlorid amónny). Ďalšie limity boli stanovené ako teplota topenia ľadu (32 ° F) a teplota ľudského tela (96 ° F, približne 2,6 ° F pod modernou hodnotou v dôsledku neskoršej redefinície stupnice). Stupnica je teraz zvyčajne definovaná dvomi pevnými bodmi: teplota, pri ktorej voda zamrzne na ľade, je definovaná ako 32 ° F a bod varu vody je definovaný ako 212 ° F. Prevodový vzorec je: $(c^{\circ}\text{C} \times 9/5) + 32 = f^{\circ}\text{F}$

Na prevod môžeme použiť upravený vzorec $C \times 1.8 + 32$.



Stlačenie tlačidla A vyvolá tú istú reakciu ako riešenie predchádzajúceho príkladu, stačí vymeniť vstupnú reakciu na `on button A pressed`.

Stlačenie tlačidla B si vyžaduje vykonať matematické operácie, ktoré nájdeme v sade príkazov `Math` (sčítanie, násobenie a zaokrúhlenie – `round`). Pre zvýšenie čitateľnosti kódu budeme používať premenné, ktoré definujeme v časti `Variables`.

Blokový jazyk je jednoduchší v tom, že sa nemusíme starať o nekonečnú slučku, ktorá čaká na stlačenie tlačidla ale priamo reagovať len na udalosti stlačenia. To v Pythone nemôžeme.

```
from microbit import *

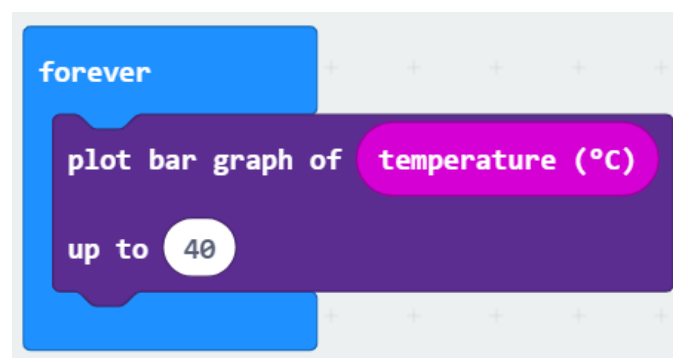
while True:
    if button_a.was_pressed():
        display.scroll(str(temperature()) + "°C ")

    if button_b.was_pressed():
        c = temperature()
        f = c * 1.8 + 32
        f = round(f)
        display.scroll(str(f) + "F")
```

Úloha 5 Vizualizuj teplotnú úroveň na displeji vo forme grafu. 40 stupňov nech je maximálna teplota na grafové zobrazenie.

V Pythone nie je štandardný príkaz na vykreslenie grafu na LED displeji. Preto ho budeme prezentovať len pre jazyk MakeCode.

Príkaz `plot bar graph of` dostane hodnotu premennej `temperature`, ktorú percentuálne prepočíta vzhľadom na vopred stanovenú maximálnu hodnotu. Preto je dôležité správne nastaviť pre akú hodnotu sa majú rozsvietiť všetky LED. Minimálna hodnota je vždy 0.



MakeCode umožňuje vizualizovať dáta ako živý graf aj cez konzolový vstup simulátora. Po spustení programu sa zobrazí tlačidlo `Show console Simulator`. Ak budeme v simulátore meniť hodnotu teploty, bude sa meniť nielen počet rozsvietených LED ale zobrazí sa aj dynamicky vykresľovaný graf, ktorého hodnoty môžeme následne stiahnuť do CSV súboru a spracovať ich aj mimo simulátora.

Po pripojení micro:bita sa zobrazí aj nové tlačidlo `Show console Device`. Po jeho stlačení sa budú vizualizovať dáta priamo z teplotného senzora na micro:bite.



Úloha 6 Miesto teploty napíš, či je zima, teplo alebo horúco. Hranice kedy pre tieto teploty si treba zvoliť podľa vlastných preferencií.

Aby sme prestriedali udalosti, vypíšeme stav po zatrasení zariadenia. Udalosť `on shake`. V MakeCode porovnanie (rovná sa, menšie, väčšie) aj vetvenie `if` nájdeme v časti `Logic`.

```
on shake ▼
  set c ▼ to temperature (°C)
  if < c ▼ < ▼ 10 > then
    show string "Zima"
  else if < c ▼ < ▼ 21 > then
    show string "Teplo"
  else
    show string "Horuco"
```

Príklad by sa dal vyriešiť aj bez použitia premennej `c`. Toto riešenie je však vhodné z dôvodu, že vo všetkých porovnaníach porovnáваме tú istú hodnotu.

```

from microbit import *

while True:
    if accelerometer.was_gestured("shake"):
        c = temperature()
        if c < 10:
            display.scroll("Zima")
        elif c < 21:
            display.scroll("Teplo")
        else:
            display.scroll("Horuco")

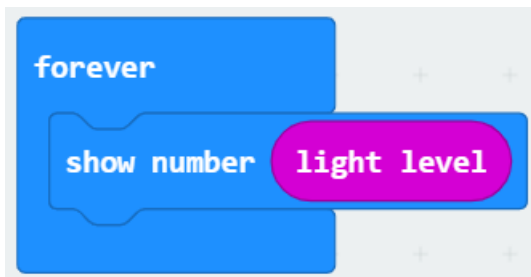
```

Nasledujúce úlohy budú zamerané na senzor merajúci intenzitu svetla, ktorý meria úroveň osvetlenia (ako jasnú alebo tmavú) miesta, kde sa nachádzame. Úroveň svetla 0 znamená tmu a 255 znamená jasné svetlo. Micro:bit v skutočnosti nemá dedikovaný svetelný senzor. V skutočnosti meria svetlo okolo neho pomocou dopadajúceho svetla na LED displej a tak je intenzita osvetlenia je len vhodnou aproximáciou svetla dopadajúceho na LEDky.

Keďže svetivosť nie je možné nastaviť ale len čítať, tak ju nájdeme v kategórii vstupov (Input). Príkaz `light_level` alebo `display.read_light_level()` len poskytne dáta aproximované z LEDiek. Opäť je možné ho vypísať, či použiť v logickom výraze alebo priradiť jeho hodnoty do premennej.

Úloha 7 Zobrazte na displeji intenzitu svetla.

Riešenie je veľmi podobné tomu, čo sme riešili v úlohe 1.



```

from microbit import *
while True:
    display.scroll(str(
display.read_light_level()))

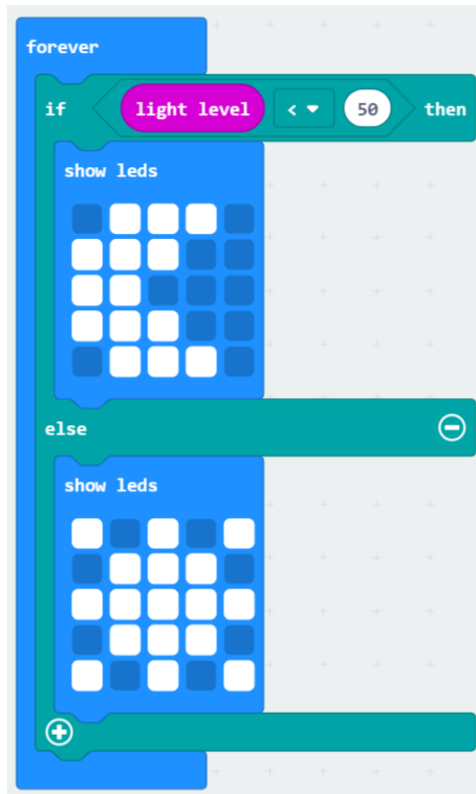
```

Pri meraní intenzity osvetlenia preto odporúčame mať vypnuté LED. Osvetlenie z diód by skreslilo výsledky.

Úloha 8 Porovnávajte intenzitu svetla a keď bude nižšia ako hodnota, ktorú si zvolíte ako zlomovú medzi dňom a nocou (napr. 50), tak zobrazte obrázok mesiaca a keď vyššia tak slnka.

Keďže treba zisťovať intenzitu svetla priebežne stále, tak použijeme nekonečnú slučku (MakeCode - `Forever`, Python - `while True:`). Vo vetvení v podmienke

použite porovnávanie aby ste zistili, či je intenzita nižšia ako zvolená hodnota. Ak je podmienka splnená tak zobrazíme mesiačik, ak nie tak zobrazíme slniečko. V MakeCode porovnávanie(rovna sa, menšie, väčšie) aj vetvenie `if` nájdete v časti `Logic`.

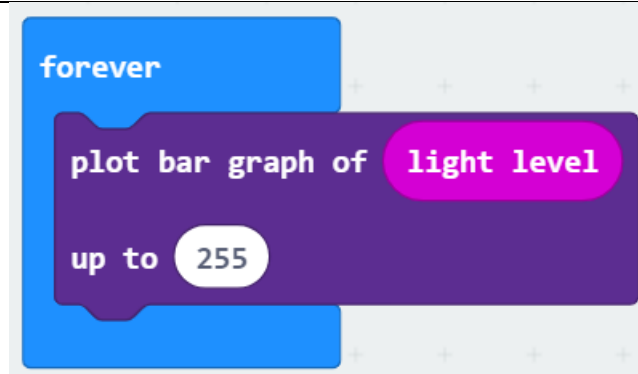


```
from microbit import *  
  
moon = Image('00990:'  
             '99900:'  
             '99000:'  
             '99900:'  
             '00990:')  
  
sun = Image('90909:'  
            '09990:'  
            '99999:'  
            '09990:'  
            '90909:')  
  
while True:  
    if display.read_light_level() < 50:  
        display.show(moon)  
    else:  
        display.show(sun)
```

Úloha 9 Vizualizuj svetelnú úroveň na displeji vo forme grafu. stupňov nech je maximálna teplota na grafové zobrazenie.

V Pythone nie je štandardný príkaz na vykreslenie grafu na LED displeji. Preto ho budeme prezentovať len pre jazyk MakeCode.

Príkaz `plot bar graph of` dostane hodnotu premennej `light level`, ktorú percentuálne prepočíta vzhľadom na vopred stanovenú maximálnu hodnotu. Preto je dôležité správne nastaviť pre akú hodnotu sa majú rozsvietiť všetky LED. Minimálna hodnota je vždy 0 a maximálna úroveň osvetlenia je 255.

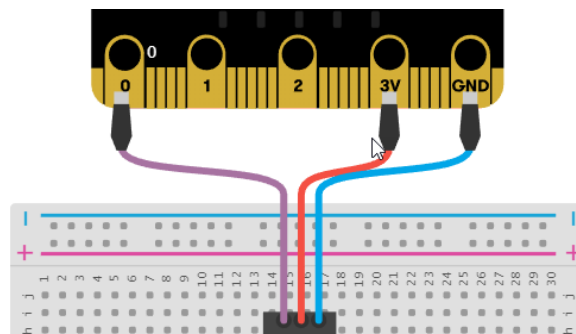


Podobne ako v úlohe 5 aj tu môžeme zobrazovať dynamický graf. V tejto úlohe môžeme dokonca zmenu osvetlenia dynamicky prezentovať aj s micro:bitom, ktorému môžeme na LED striedavo svietiť baterkou a zatemniť displej rukou.

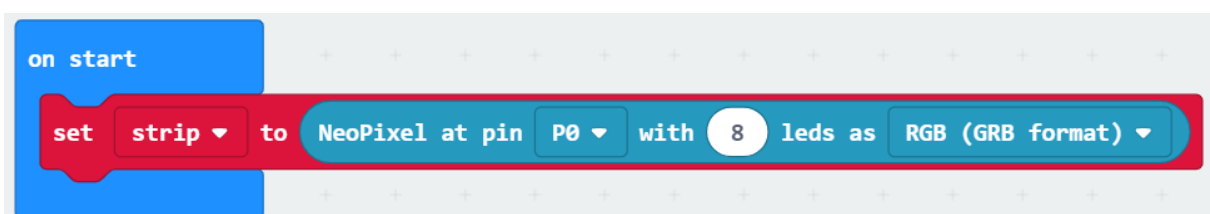
Úloha 10 Pripojte k micro:bitovi LED pásik. Rozsviette ho, keď sa zotmie, zhasnite keď začne svietiť slnko, čiže simulujte automatické zažínanie a vypínanie osvetlenia.

Začnime najskôr pripojením LED pásika. Použijeme pásik Adafruit Neopixel LED strip. Ak ho chceme programovať, musíme pridať množinu príkazov do prostredia MakeCode. Vyberieme ponuku **+** Extensions . A z ponúkaných možností rozšírenia vyberieme možnosť **Neopixel**.

Následne pripojíme pásik pomocou krokosvoriek k micro:bitovi. Budeme postupovať podľa obrázku nižšie.

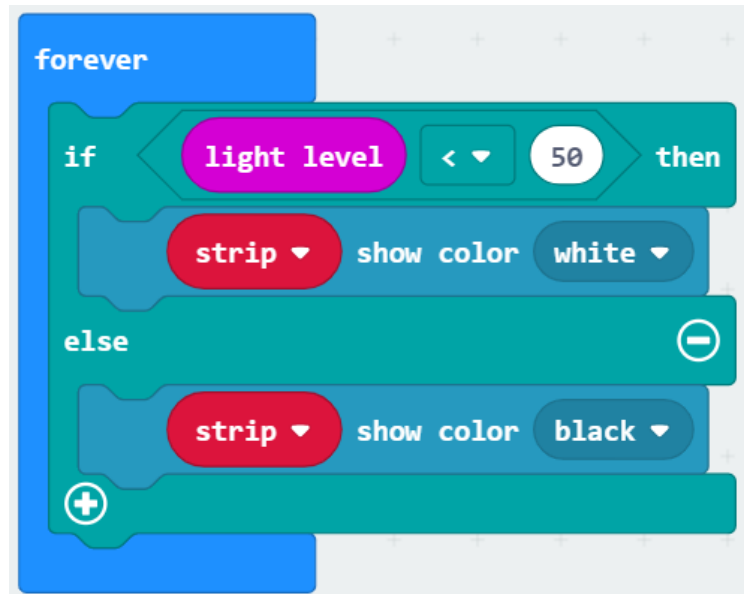


Najskôr musíme nastaviť správne parametre pre pásik. Toto je vhodné urobiť pri zapnutí micro:bita. Keďže sme ho pripojili na dátový PIN s označením 0, vyberieme z ponuky práve tento PIN. Následne určíme koľko LEDiek obsahuje pásik. Naše vzorové riešenie je pre pásik o veľkosti 8 LED.



Pásik máme nastavený a môžeme prísť k meraniu intenzity osvetlenia, ktorú je vhodné merať v nekonečnej slučke. Ak nájdeme hodnotu nižšiu ako 50 (z intervalu 0 .. 255), tak budeme považovať, že je už noc a zažneme osvetlenie. Celý pásik bude

svietiť bielou farbou. Ak svetelnosť vystúpi nad túto hranicu, pásik sa vypne, čo je identické s nastavením LED na čiernu farbu.



Vlastnosti pásika nastavíme v konštruktore `neopixel.NeoPixel(pin0, 8)`. V Pythone neexistuje možnosť naraz rozsvietiť všetky LED. Pásik sa programuje ako list, kde každá LED má svoj index. Preto musíme prejsť všetky LEDky a postupne im nastaviť konkrétnu farbu. Čierna farba je v RGB zápise (0,0,0). Biela farba je v RGB zápise (255, 255, 255). Príkazom `strip[] =` nastavíme farby len v pamäti micro:bita. Ak ich chcem rozsvietiť musíme použiť príkaz `strip.show()`.

```
from microbit import *
import neopixel

strip = neopixel.NeoPixel(pin0, 8)

while True:
    if display.read_light_level() < 50:
        for i in range(8):
            strip[i] = (255, 255, 255)
        strip.show()
    else:
        for i in range(8):
            strip[i] = (0, 0, 0)
        strip.show()
```