1. Čarobni štapić (Detektor pokreta s animacijom) ✨

- **Moduli:** Senzor pokreta (PIR) -> pin **P0**
- **Opis:** Kada zamahneš rukom ispred senzora, micro:bit se "budi" i na ekranu prikazuje animaciju srca koje kuca.

```
from microbit import *


while True:
    if pin0.read_digital() == 1:
        # Animacija kucanja srca
        display.show(Image.HEART)
        sleep(300)
        display.show(Image.HEART_SMALL)
        sleep(300)
    else:
        display.clear()  # Ekran je ugašen dok nema pokreta
    sleep(100)
```

2. Kameleon svjetlo (Promjena boje puhanjem)

- **Moduli:** Senzor zvuka -> pin **P0**, RGB LED traka -> pin **P1**
- **Opis:** Kada jako puhneš u senzor zvuka, micro:bit nasumično promijeni boju na RGB traci. Svaki jači zvuk donosi novu boju!

```
from microbit import *

import random


while True:

    # Ako je zvuk glasan (pohvala/pljesak/puhanje)

    if pin0.read_analog() > 600:

        # Nasumično biramo sliku/boju (ovisi o verziji trake)

        # Za standardni micro:bit ekran mijenjamo nasumične sličice

        slicice = [Image.HAPPY, Image.SAD, Image.GHOST, Image.PACMAN]

        display.show(random.choice(slicice))

        sleep(500)

    sleep(50)
```

4. Igra "Brzi prsti" (Refleksni test) ⏱️

- **Moduli:** LED modul -> pin **P1**, Gumb -> pin **P2**
- **Opis:** LED svjetlo se upali u nasumičnom trenutku. Tvoj zadatak je stisnuti gumb što brže možeš nakon što se svjetlo upali. Ako uspiješ, ekran se nasmije!

```
from microbit import *

import random


while True:

    display.clear()

    pin1.write_digital(0)


    # Čekaj nasumično između 2 i 5 sekundi

    sleep(random.randint(2000, 5000))


    pin1.write_digital(1)  # Upali LED - kreni!


    # Čekaj reakciju igrača (maksimalno 1 sekundu)

    for i in range(100):

        if pin2.read_digital() == 1:

            display.show(Image.HAPPY)  # Pobijedio si!

            sleep(1000)

            break

        sleep(10)
```

5. Elektronička kockica za društvene igre 🎲

- **Moduli:** Gumb -> pin **P0**
- **Opis:** Umjesto bacanja plastične kockice za Čovječe ne ljuti se, stisni Boson gumb i micro:bit će ti na ekranu prikazati nasumičan broj od 1 do 6.

```
from microbit import *

import random


while True:

    if pin0.read_digital() == 1:

        # Efekt vrtenja kockice na ekranu

        for _ in range(5):

            display.show(random.randint(1, 6))

            sleep(100)

        # Konačni broj

        broj = random.randint(1, 6)

        display.show(broj)

        sleep(2000)  # Prikazuj broj 2 sekunde

    sleep(50)
```

6. Strašna kutija (Halloween fora) 🎃

- **Moduli:** Senzor pokreta -> pin **P1**, Zujalica -> pin **P8**
- **Opis:** Sakrij kutiju. Kada joj se netko približi, senzor pokreta aktivira zujalicu koja ispušta jezivi opadajući ton, a na ekranu se pojavi duh.

```
from microbit import *

import music


while True:

    if pin1.read_digital() == 1:

        display.show(Image.GHOST)

        # Padajući jezivi ton

        for f in range(800, 200, -20):

            music.pitch(f, duration=20, pin=pin8)

        display.clear()

    sleep(100)
```

7. Glazbena kutija za nakit 🎜

- **Moduli:** Senzor svjetla -> pin **P0**, Zujalica -> pin **P8**
- **Opis:** Kada je kutija zatvorena, unutra je mrak i sve miruje. Čim netko otvori kutiju, svjetlo ulazi unutra, senzor to prepoznaje i zujalica počinje svirati sretnu melodiju.

```
from microbit import *

import music


while True:

    # Ako ima svjetla (kutija je otvorena)

    if pin0.read_analog() > 400:

        # Sviraj ugrađenu melodiju (npr. Oda radosti ili rođendanska)

        music.play(music.ENTERTAINER, pin=pin8)

    else:

        music.stop(pin=pin8)

    sleep(500)
```

8. Indikator raspoloženja (Semafor emocija) 😊

- **Moduli:** Potenciometar -> pin **P0**
- **Opis:** Okretanjem gumba (potenciometra) mijenjaš raspoloženje micro:bita na ekranu. Na početku je tužan, u sredini ravnodušan, a na kraju sretan!

```
from microbit import *

while True:
    vrijednost = pin0.read_analog()

    if vrijednost < 300:
        display.show(Image.SAD)
    elif vrijednost >= 300 and vrijednost < 700:
        display.show(Image.MEH)
    else:
        display.show(Image.HAPPY)

    sleep(100)
```

9. Čarobni pljesak (Senzor zvuka pali/gasi svjetlo) 👏

- **Moduli:** Senzor zvuka -> pin **P0**, LED modul -> pin **P2**
- **Opis:** Baš kao u futurističkim filmovima – jednim jakim pljeskom upali svjetlo u sobi, a idućim jakim pljeskom ga ugas

```
from microbit import *


svjetlo_upaljeno = False


while True:
    # Ako detektira glasan zvuk (pljesak)
    if pin0.read_analog() > 700:
        svjetlo_upaljeno = not svjetlo_upaljeno  # Obratno od trenutnog stanja


        if svjetlo_upaljeno:
            pin2.write_digital(1)
        else:
            pin2.write_digital(0)


        sleep(500)  # Pauza da priguši odjek pljeska
    sleep(50)
```

10. Detektor laži (Senzor dodira / Vlage) 😉

- **Moduli:** Senzor dodira (ili vlage kože) -> pin **P0**, LED modul -> pin **P1**
- **Opis:** Kada smo nervozni ili lažemo, prsti nam se lagano znoje. Postavi prst na senzor. Ako očitana vrijednost naglo skoči, LED modul će početi brzo bljeskati – uhvaćen u laži!

```
from microbit import *


while True:
    vlaga_koze = pin0.read_analog()


    # Ako je senzor jako osjetljiv na dodir prsta
    if vlaga_koze > 400:
        display.show(Image.ANGRY)
        pin1.write_digital(1)
        sleep(100)
        pin1.write_digital(0)
        sleep(100)
    else:
        display.show(Image.HAPPY)
        pin1.write_digital(0)
    sleep(100)
```

11. "Vrući krumpir" (Zabavna igra za društvo) 🥔

- **Moduli:** Gumb -> pin **P0**, Zujalica -> pin **P8**
- **Opis:** Igrači sjede u krugu i dodaju micro:bit. Kada stisneš gumb, zujalica počne kucati sve brže i brže u nasumičnom trajanju. Kod koga zujalica neprekidno zasvira "BEEP" – taj je izgubio!

```
from microbit import *

import music

import random


while True:

    display.show(Image.SURPRISED)

    if pin0.read_digital() == 1:  # Igra kreće pritiskanjem gumba

        trajanje_igre = random.randint(10, 30)


        for i in range(trajanje_igre):

            music.pitch(600, duration=100, pin=pin8)

            # Kako igra ide kraju, pauza između kucanja je sve kraća

            sleep(500 - (i * 10))


        # Kraj igre - eksplozija krumpira!

        display.show(Image.SKULL)

        music.play(music.WAW_WAW_WAW, pin=pin8)

        sleep(3000)
```

12. Policijska sirena (Brza izmjena boja) 🚓

- **Moduli:** RGB traka -> pin **P0**
- **Opis:** Polovica trake svijetli jarko plavo, a druga polovica jarko crveno. Zatim u ritmu prave sirene zamijene mjesta. [1, 2]

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)


while True:

    # Prva kombinacija: prve 3 lampice plave, ostale crvene

    for i in range(0, 3):

        traka[i] = (0, 0, 255)   # Plava (R, G, B)

    for i in range(3, 7):

        traka[i] = (255, 0, 0)   # Crvena

    traka.show()

    sleep(200)


    # Druga kombinacija: zamjena mjesta

    for i in range(0, 3):

        traka[i] = (255, 0, 0)   # Crvena

    for i in range(3, 7):

        traka[i] = (0, 0, 255)   # Plava

    traka.show()

    sleep(200)
```

13. Svjetlosni termometar (Sobni ugođaj) 🌡️

- **Moduli:** RGB traka -> pin **P0**
- **Opis:** Koristi ugrađeni senzor temperature u micro:bitu. Ako je u sobi hladno (ispod 22°C), cijela traka svijetli plavo. Ako je ugodno (22°C - 26°C), svijetli zeleno, a ako je vruće (iznad 26°C), svijetli crveno. [1, 2, 3]

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)


while True:

    temp = temperature() # Očitavanje temperature


    if temp < 22:

        boja = (0, 0, 255)    # Plava za hladno

    elif temp >= 22 and temp <= 26:

        boja = (0, 255, 0)    # Zelena za ugodno

    else:

        boja = (255, 0, 0)    # Crvena za vruće


    # Oboji sve lampice na traci u odabranu boju

    for i in range(7):

        traka[i] = boja

    traka.show()


    sleep(1000) # Provjera svake sekunde
```

14. Laserski mač iz Star Warsa (Glatko paljenje) ⚔️

- **Moduli:** Gumb -> pin **P1**, RGB traka -> pin **P0**
- **Opis:** Kada stisneš gumb, lampice se ne upale odjednom, nego se pale jedna po jedna u nizu, stvarajući vizualni efekt izvlačenja laserskog mača (poput onog koji koristi Yoda - zeleni). Kada ponovno stisneš, gase se obrnutim redoslijedom. [1, 2]

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)

mac_upaljen = False


while True:

  if pin1.read_digital() == 1:

    mac_upaljen = not mac_upaljen


    if mac_upaljen:

      # Pali se jedna po jedna odozdo prema gore

      for i in range(7):

        traka[i] = (0, 255, 0) # Zelena boja mača

        traka.show()

        sleep(100)

    else:

      # Gasi se jedna po jedna odozgo prema dolje

      for i in range(6, -1, -1):

        traka[i] = (0, 0, 0) # Ugašeno

        traka.show()
```

```
        sleep(100)


    sleep(500) # Sprječava titranje gumba
sleep(50)
```

15. Svjetlosni "Volume bar" (Mjerač buke) 🔊

- **Moduli:** Senzor zvuka -> pin **P1**, RGB traka -> pin **P0**
- **Opis:** Što glasnije vičeš ili plješćeš, to se više lampica na traci upali (isto kao indikator glasnoće na zvučnicima). Kada zavlada tišina, lampice se spuštaju na nulu. [1, 2]

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)


while True:

    buka = pin1.read_analog() # Vrijednosti od 0 do 1023


    # Pretvaramo raspon buke u broj lampica (od 0 do 7)

    # Primjerice, svakih ~130 bodova buke pali novu lampicu

    broj_lampica = int(buka / 130)


    # Oboji traku ovisno o glasnoći

    for i in range(7):

        if i < broj_lampica:

            traka[i] = (255, 255, 0) # Žuta boja za aktivne lampice

        else:

            traka[i] = (0, 0, 0)     # Ugašeno za ostale


    traka.show()

    sleep(30) # Brzo osvježavanje za gladak efekt
```

16. Igra "Uhvati zeleno svjetlo" (Zabavni arkadni test) 🎮

- **Moduli:** Gumb -> pin **P1**, RGB traka -> pin **P0**
- **Opis:** Jedna crvena lampica "trči" brzo s jednog kraja trake na drugi i tako ukrug. Srednja lampica (broj 3) je ciljna zona. Tvoj zadatak je stisnuti gumb točno u milisekundi kada crvena lampica stigne na sredinu. Ako pogodiš, cijela traka bljesne zeleno! [1, 2, 3]

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)

pozicija = 0


while True:

    # Resetiraj cijelu traku u crno (ugašeno)

    for i in range(7):

        traka[i] = (0, 0, 0)


    # Označi središnju "ciljnu" lampu laganom plavom bojom

    traka[3] = (0, 0, 20)


    # Postavi trenutnu "trčeću" lampicu u crvenu boju

    traka[pozicija] = (255, 0, 0)

    traka.show()


    # Provjera je li igrač stisnuo gumb u ovom koraku

    if pin1.read_digital() == 1:

        if pozicija == 3: # Pogodak!
```

```
        for _ in range(3): # Bljesni zeleno 3 puta za pobjedu
            for k in range(7): traka[k] = (0, 255, 0)
            traka.show()
            sleep(150)
            for k in range(7): traka[k] = (0, 0, 0)
            traka.show()
            sleep(150)
    else: # Promašaj!
        for k in range(7): traka[k] = (255, 0, 0) # Bljesni crveno
        traka.show()
        sleep(500)

# Pomakni lampicu na iduće mjesto
pozicija = (pozicija + 1) % 7
sleep(150) # Brzina trčanja svjetla (smanji broj za veću težinu!)
```

16. RGB Ručni mješač boja (Duga na dlanu) 🎨

- **Moduli:** RGB traka -> **P0**, Potenciometar -> **P1**, Gumb -> **P2**
- **Opis:** Okretanjem gumba biraš jednu od 7 čistih boja (crvena, narančasta, žuta, zelena, plava, indigo, ljubičasta). Kada izabereš savršenu boju, stisni Boson gumb (P2) kako bi cijela traka bljesnula tom bojom uz kratki zvučni signal potvrde.

```
from microbit import *

import neopixel

import music


traka = neopixel.NeoPixel(pin0, 7)


# Popis 7 duginih boja (R, G, B)

boje = [

    (255, 0, 0),     # Crvena

    (255, 127, 0),   # Narančasta

    (255, 255, 0),   # Žuta

    (0, 255, 0),     # Zelena

    (0, 0, 255),     # Plava

    (75, 0, 130),    # Indigo

    (148, 0, 211)    # Ljubičasta

]


while True:

    vrijednost = pin1.read_analog()

    # Mapiramo vrijednost 0-1023 u indeks od 0 do 6

    indeks = int(vrijednost / 147)
```

```
if indeks > 6: indeks = 6

# Prikaži odabranu boju samo na PRVOJ lampici (kao pretpregled)
for i in range(7):
    traka[i] = (0, 0, 0)
traka[0] = boje[indeks]
traka.show()

# Ako stisneš gumb, oboji CIJELU traku i sviraj ton
if pin2.read_digital() == 1:
    for i in range(7):
        traka[i] = boje[indeks]
    traka.show()
    music.pitch(600, duration=150, pin=pin8)
    sleep(500)

sleep(50)
```

17. Svjetlosni trezor (Otključavanje šifre) 🔒🎵

- **Moduli:** RGB traka -> **P0**, Potenciometar -> **P1**, Zujalica -> **P8**
- **Opis:** Zamisli da je potenciometar kotačić na sefu. Tajna šifra za otključavanje je točno na sredini (vrijednost između 500 i 530). Kako se približavaš točnom broju, lampice na traci se pale jedna po jedna (od krajeva prema sredini) i zujalica kuca sve brže. Kad pogodiš sredinu, traka svijetli zeleno i sef svira pobjednički ton!

```
from microbit import *

import neopixel

import music


traka = neopixel.NeoPixel(pin0, 7)


while True:

    pozicija = pin1.read_analog()


    # Izračunavamo koliko smo blizu cilju (512)

    udaljenost = abs(512 - pozicija)


    # Ugasi traku na početku kruga

    for i in range(7): traka[i] = (0, 0, 0)


    if udaljenost < 20:  # SEF OTKLJUČAN!

        for i in range(7): traka[i] = (0, 255, 0) # Sve zeleno

        traka.show()

        music.play(music.CHIME, pin=pin8)

        sleep(2000)

    else:
```

```
# Što smo bliže, to se više lampica pali prema sredini
broj_lampica = int((1023 - udaljenost) / 146)
for i in range(broj_lampica):
    if i < 7: traka[i] = (255, 0, 0) # Crveni indikator blizine
traka.show()

# Zvuk kucanja ovisi o blizini cilja
music.pitch(400, duration=20, pin=pin8, wait=False)
sleep(int(udaljenost / 2) + 10)
```

18. "DJ Pitcher" (Brzina svjetlosnog vala i zvuka) 🕺🎛️

- **Moduli:** RGB traka -> **P0**, Potenciometar -> **P1**, Zujalica -> **P8**
- **Opis:** Napravi svoj DJ pult. Jedna lampica stalno putuje niz traku naprijed-natrag i ostavlja "rep" u boji. Okretanjem potenciometra uživo ubrzavaš ili usporavaš to svjetlo, a zujalica istovremeno mijenja visinu tona prateći ritam tvoje ruke.

```
from microbit import *

import neopixel

import music


traka = neopixel.NeoPixel(pin0, 7)

smjer = 1

led = 0


while True:

    gumb_val = pin1.read_analog()


    # Prilagođavamo brzinu (pauzu) i visinu tona pomoću gumba

    pauza = int(20 + (gumb_val * 0.3))    # Pauza od 20ms do 320ms

    ton = int(200 + (gumb_val * 0.8))     # Frekvencija od 200Hz do 1000Hz


    # Efekt brisanja s laganim blijeđenjem

    for i in range(7): traka[i] = (0, 0, 0)


    traka[led] = (0, 255, 255) # Cijan boja za trkača

    traka.show()
```

```
music.pitch(ton, duration=20, pin=pin8, wait=False)


# Pomakni LED lampicu
led += smjer
if led == 6 or led == 0:
    smjer = -smjer # Promijeni smjer na rubovima trake


sleep(pauza)
```

19. Svjetlosni pješčani sat s upozorenjem (Timer za učenje) ⏳⏰

- **Moduli:** RGB traka -> **P0**, Potenciometar -> **P1**, Zujalica -> **P8**
- **Opis:** Okretanjem gumba postavljaš vrijeme na tajmeru (od 1 do 7 lampica, svaka lampica predstavlja npr. 5 sekundi za brzi test). Kada pustiš gumb, traka se počinje prazniti – lampice se gase jedna po jedna. Kada se ugasi zadnja lampica, aktivira se glasan alarm za kraj vremena!

```
from microbit import *

import neopixel

import music


traka = neopixel.NeoPixel(pin0, 7)


# 1. FAZA: Postavljanje vremena pomoću potenciometra

display.show("P") # "P" za postavljanje

while True:

   vrijednost = pin1.read_analog()

   broj_lampica = int(vrijednost / 147) + 1 # Koliko lampica svijetli (1-7)


   for i in range(7):

      if i < broj_lampica: traka[i] = (0, 0, 255) # Plave lampice vremena

      else: traka[i] = (0, 0, 0)

   traka.show()


   # Ako makneš ruku i gumb miruje na nekoj poziciji, igra kreće za 2 sekunde

   sleep(100)

   if abs(pin1.read_analog() - vrijednost) < 5:

      break
```

```
# 2. FAZA: Odbrojavanje vremena
display.show("GO")
for preostalo in range(broj_lampica - 1, -1, -1):
    sleep(3000) # Svaka lampica traje 3 sekunde (promijeni na više za pravi tajmer)
    traka[preostalo] = (0, 0, 0) # Ugasi jednu lampicu
    traka.show()
    music.pitch(500, duration=100, pin=pin8) # Kratki "bip" za sekundu

# 3. FAZA: KRAJ VREMENA (Alarm)
display.show(Image.SKULL)
for _ in range(5):
    for i in range(7): traka[i] = (255, 0, 0) # Bljeska crveno
    traka.show()
    music.play(music.BA_DING, pin=pin8)
    for i in range(7): traka[i] = (0, 0, 0)
    traka.show()
    sleep(200)
```

20. Simulator disanja svemirskog broda (Glatko prigušivanje - Dimmer) 🚀

- **Moduli:** RGB traka -> **P0**, Potenciometar -> **P1**
- **Opis:** Ova vježba te uči kako upravljati jačinom (svjetlinom) RGB trake. Okretanjem potenciometra ne mijenjaš broj lampica, već cijela traka glatko mijenja intenzitet boje od jedva vidljive do super jake svjetlosti. Savršeno za izradu ambijentalnog svjetla za sobu!

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)


while True:

    vrijednost_gumba = pin1.read_analog() # Vrijednost od 0 do 1023


    # Budući da jačina boje u RGB-u ide maksimalno do 255,

    # dijelimo vrijednost s 4 kako bismo dobili ispravan raspon.

    jacina = int(vrijednost_gumba / 4)


    # Postavi sve lampice na ljubičastu boju, ali s prilagođenom jačinom

    for i in range(7):

        traka[i] = (jacina, 0, jacina) # Miješanjem Crvene i Plave nastaje ljubičasta


    traka.show()

    sleep(50) # Brzi odaziv na okretanje gumba
```

21. (Najlakša): Klik-Blink preko radija

Ovo je osnovni "Hello World" radio komunikacije. Kada stisneš gumb na jednom micro:bitu, na drugom se upali RGB traka. Nema provjere teksta, samo hvatamo bilo kakav signal.

Kod za ODAŠILJAČ (Daljinski):

```
from microbit import *

import radio


radio.on()

radio.config(channel=10) # Oba uređaja moraju biti na istom kanalu


while True:

    if button_a.was_pressed():

        radio.send("1")  # Šalje najjednostavniji signal "1"

    sleep(50)
```

Kod za PRIJAMNIK (Baza s RGB trakom na P0):

```
from microbit import *

import radio

import neopixel


radio.on()

radio.config(channel=10)

traka = neopixel.NeoPixel(pin0, 7)


while True:

    poruka = radio.receive()
```

```
if poruka == "1": # Ako uhvati signal "1"
    # Upali sve lampice u crvenu boju
    for i in range(7):
        traka[i] = (255, 0, 0)
    traka.show()
    sleep(2000) # Svijetli 2 sekunde

    # Ugasi traku
    for i in range(7):
        traka[i] = (0, 0, 0)
    traka.show()

sleep(20)
```

22. (Lakša): Prekidač za svjetlo (ON / OFF)

Sada radimo pravi bežični prekidač. Gumb A pali svjetlo i ono ostaje upaljeno. Svjetlo gasiš tek kada stisneš gumb B na daljinskom.

Kod za ODAŠILJAČ (Daljinski):

```
from microbit import *
import radio

radio.on()
radio.config(channel=10)

while True:
    if button_a.was_pressed():
        radio.send("upali")
    if button_b.was_pressed():
        radio.send("ugasi")
    sleep(50)
```

Kod za PRIJAMNIK (Baza s RGB trakom na P0):

```
from microbit import *
import radio
import neopixel

radio.on()
radio.config(channel=10)
traka = neopixel.NeoPixel(pin0, 7)

while True:
```

```
poruka = radio.receive()

if poruka == "upali":
    for i in range(7):
        traka[i] = (0, 255, 0) # Upali zelenu boju
    traka.show()

elif poruka == "ugasi":
    for i in range(7):
        traka[i] = (0, 0, 0) # Ugasi sve
    traka.show()

sleep(20)
```

23. (Srednja): Biranje boja na daljinu + Zvuk 🎨🎵

Idemo korak dalje. Daljinski sada upravlja bojama i zvukom baze. Svaki pritisak na gumb A mijenja boju na traci, a prijamnik uz promjenu boje odsvira i kratki ton na zujalici.

Kod za ODAŠILJAČ (Daljinski):

```
from microbit import *

import radio


radio.on()

radio.config(channel=10)

brojac = 0


while True:

    if button_a.was_pressed():

        brojac = (brojac + 1) % 3 # Kruži kroz brojeve 0, 1, 2


        if brojac == 0:

            radio.send("crvena")

        elif brojac == 1:

            radio.send("plava")

        elif brojac == 2:

            radio.send("zelena")


    sleep(50)
```

Kod za PRIJAMNIK (Baza s RGB trakom na P0 i Zujalicom na P8):

```
from microbit import *
import radio
import neopixel
import music

radio.on()
radio.config(channel=10)
traka = neopixel.NeoPixel(pin0, 7)

while True:
    poruka = radio.receive()

    if poruka:
        # Odredi boju ovisno o poruci
        if poruka == "crvena":
            boja = (255, 0, 0)
            ton = 440 # Nota A
        elif poruka == "plava":
            boja = (0, 0, 255)
            ton = 554 # Nota C#
        elif poruka == "zelena":
            boja = (0, 255, 0)
            ton = 659 # Nota E

        # Oboji traku i pusti zvuk
```

```
    for i in range(7):
        traka[i] = boja
    traka.show()

    music.pitch(ton, duration=150, pin=pin8)

sleep(20)
```

24. ODAŠILJAČ (Senzor na vratima)

Ovaj micro:bit stavljaš na bateriju i zalijepiš ga (ili nasloniš) na vrata sobe ili ladicu. Koristi svoj ugrađeni senzor pokreta (akcelerometar). Čim netko otvori vrata i protrese ga, on bežično šalje signal "alarm".

Kod za Odašiljač:

```
from microbit import *

import radio


# Uključi radio i postavi kanal 10

radio.on()

radio.config(channel=10)


display.show(Image.ASLEEP) # Prikazuje ikonu "spavanje" dok čuva stražu


while True:

    # Ako netko pomakne ili protrese micro:bit (vrata se otvaraju)

    if accelerometer.was_gesture("shake"):

        display.show(Image.SURPRISED) # Kratko se iznenadi

        radio.send("alarm")          # Šalje signal za uzbunu kroz zrak

        sleep(2000)                  # Pauza od 2 sekunde da ne šalje previše signala odjednom

        display.show(Image.ASLEEP)


    sleep(100)
```

Uređaj 2: PRIJAMNIK (Centralni alarm s RGB trakom)

Ovaj micro:bit spoji na Boson ploču, a **RGB traku spoji na pin P0**. Kada primi poruku "alarm", aktivira se jarka policijska rasvjeta na traci, na njegovu ekranu se pojavi znak lubanje, a ugrađeni zvučnik počinje glasno svirati brzu dvotonsku sirenu.

Kod za Prijamnik:

```
from microbit import *

import radio

import neopixel

import music


# Uključi radio na istom kanalu 10

radio.on()

radio.config(channel=10)


# Pripremi RGB traku s 7 lampica na pinu P0

traka = neopixel.NeoPixel(pin0, 7)


display.show(Image.HAPPY) # Sve je u redu, sustav je aktivan


while True:

    poruka = radio.receive()


    # Ako kroz zrak stigne poruka "alarm"

    if poruka == "alarm":

        display.show(Image.SKULL) # Prikaži lubanju na ekranu micro:bita


        # Pokreni sirenu i bljeskanje koje traje 5 krugova
```

```
    for _ in range(5):
        # 1. korak: Plavo svjetlo + visoki ton iz ugrađenog zvučnika
        for i in range(7):
            traka[i] = (0, 0, 255)
        traka.show()
        music.pitch(880, duration=150) # Visoki ton (880 Hz)

        # 2. korak: Crveno svjetlo + niži ton iz ugrađenog zvučnika
        for i in range(7):
            traka[i] = (255, 0, 0)
        traka.show()
        music.pitch(660, duration=150) # Niži ton (660 Hz)

    # Nakon što alarm završi, ugasi traku i vrati sretno lice
    for i in range(7):
        traka[i] = (0, 0, 0)
    traka.show()
    display.show(Image.HAPPY)

sleep(20)
```

25. Vježba: Bežični "Pager" (Slanje brzih poruka) 💬

Ova vježba simulira stari uređaj za slanje kratkih poruka (pager). Stiskom na gumbe odašiljača šalješ točno definirane riječi, a prijamnik ih ispisuje na ekranu i pali svjetlo u boji te poruke kako bi privukao pažnju.

ODAŠILJAČ (Daljinski):

```
from microbit import *

import radio


radio.on()

radio.config(channel=25)


while True:

    if button_a.was_pressed():

        radio.send("POMOC")  # Šalje poruku za pomoć

        display.scroll("Poslano: POMOC")

    if button_b.was_pressed():

        radio.send("BOK")    # Šalje prijateljski pozdrav

        display.scroll("Poslano: BOK")

    sleep(50)
```

PRIJAMNIK (Baza s RGB trakom na P0):

```
from microbit import *

import radio

import neopixel


radio.on()
```

```
radio.config(channel=25)
traka = neopixel.NeoPixel(pin0, 7)

while True:
    poruka = radio.receive()

    if poruka:
        # Ako stigne poruka "POMOC" - svijetli crveno i ispiši tekst
        if poruka == "POMOC":
            for i in range(7): traka[i] = (255, 0, 0)
            traka.show()
            display.scroll("HITNO: POMOC!")

        # Ako stigne poruka "BOK" - svijetli zeleno i ispiši tekst
        elif poruka == "BOK":
            for i in range(7): traka[i] = (0, 255, 0)
            traka.show()
            display.scroll("Prijatelj kaze: BOK!")

        # Ugasi traku nakon što poruka prođe ekranom
        for i in range(7): traka[i] = (0, 0, 0)
        traka.show()

    sleep(20)
```

26. Vježba: Daljinski birač svjetlosnih efekata (Mode Selector)

Umjesto slanja običnih riječi, odašiljač šalje nazive svjetlosnih efekata: "DUGA", "POLICIJA" ili "UGASI". Prijamnik prepoznaje tekstualnu poruku i pokreće cijelu petlju za taj specifični animirani efekt na RGB traci.

ODAŠILJAČ (Daljinski):

```
from microbit import *

import radio


radio.on()

radio.config(channel=25)

efekt_id = 0


while True:

    # Prikaz trenutnog načina rada na odašiljaču

    if efekt_id == 0: display.show("1") # Duga

    elif efekt_id == 1: display.show("2") # Policija

    elif efekt_id == 2: display.show("0") # Ugasi


    # Gumb A mijenja efekt koji želiš poslati

    if button_a.was_pressed():

        efekt_id = (efekt_id + 1) % 3


    # Gumb B šalje točan tekstualni naziv efekta preko radija

    if button_b.was_pressed():

        if efekt_id == 0: radio.send("DUGA")
```

```
        elif efekt_id == 1: radio.send("POLICIJA")
        elif efekt_id == 2: radio.send("UGASI")
        display.scroll("Poslano!")

    sleep(50)
```

PRIJAMNIK (Baza s RGB trakom na P0):

```
from microbit import *
import radio
import neopixel

radio.on()
radio.config(channel=25)
traka = neopixel.NeoPixel(pin0, 7)

trenutni_efekt = "UGASI"

while True:
    # Prvo provjeri je li stigao novi tekst s radija
    poruka = radio.receive()
    if poruka:
        trenutni_efekt = poruka  # Zapamti novi efekt (DUGA, POLICIJA ili UGASI)

    # --- IZVOĐENJE EFEKATA NA RGB TRAKI ---
    if trenutni_efekt == "DUGA":
        # Svaka lampica dobiva drugu boju
```

```
        dugine_boje = [(255,0,0), (255,127,0), (255,255,0), (0,255,0), (0,0,255), (75,0,130),
(148,0,211)]
        for i in range(7):
            traka[i] = dugine_boje[i]
        traka.show()

    elif trenutni_efekt == "POLICIJA":
        # Brzo bljeskanje crveno-plavo
        for i in range(7): traka[i] = (0, 0, 255) if i < 4 else (255, 0, 0)
        traka.show()
        sleep(100)
        for i in range(7): traka[i] = (255, 0, 0) if i < 4 else (0, 0, 255)
        traka.show()
        sleep(100)

    elif trenutni_efekt == "UGASI":
        for i in range(7): traka[i] = (0, 0, 0)
        traka.show()

    sleep(20) # Kratka pauza u glavnoj petlji
```

Ova vježba simulira **bežični meteorološki sustav**.

Jedan micro:bit se nalazi vani (ili na suncu) i mjeri temperaturu pomoću svog ugrađenog senzora, a zatim tu vrijednost bežično šalje unutra na drugi micro:bit. Drugi micro:bit prima podatak, ispisuje točnu temperaturu na svom ekranu i pali **RGB traku na pinu P0** u odgovarajućoj boji:

Uređaj 1: ODAŠILJAČ (Senzor vani)

Ovaj micro:bit stavljaš na bateriju i on svake 2 sekunde očita temperaturu te je šalje kroz zrak kao tekstualnu poruku.

```
from microbit import *

import radio


# Uključi radio i postavi kanal 30

radio.on()

radio.config(channel=30)


while True:

    # Očitaj trenutnu temperaturu s ugrađenog senzora micro:bita

    trenutna_temp = temperature()


    # Kratko prikaži izmjereni broj na ekranu odašiljača

    display.show(trenutna_temp)


    # Šaljemo temperaturu kao tekst (moramo broj pretvoriti u tekst pomoću str)

    radio.send(str(trenutna_temp))


    # Pričekaj 2 sekunde prije idućeg mjerenja i slanja

    sleep(2000)
```

Uređaj 2: PRIJAMNIK (Sobna baza s RGB trakom na pinu P0)

Ovaj micro:bit je spojen na tvoju Boson ploču. On stalno sluša radio, pretvara primljeni tekst natrag u broj i upravlja RGB trakom i ekranom.

```
from microbit import *

import radio

import neopixel


# Uključi radio na istom kanalu 30

radio.on()

radio.config(channel=30)


# Pripremi RGB traku s 7 lampica na pinu P0

traka = neopixel.NeoPixel(pin0, 7)


# Početno gašenje trake

for i in range(7): traka[i] = (0, 0, 0)

traka.show()


while True:

    # Provjeri je li stigla poruka s temperaturom

    poruka = radio.receive()


    if poruka:

        # Pretvaramo primljeni tekst natrag u cijeli broj (int)

        temp = int(poruka)
```

```
    # Ispiši primljenu temperaturu na ekranu baze (npr. "24C")
    display.scroll(str(temp) + "C")

    # --- ODREĐIVANJE BOJE RGB TRAKE ZAVISNO O TEMPERATURI ---
    if temp < 22:
        boja = (0, 0, 255)     # PLAVA za hladno
    elif temp >= 22 and temp <= 26:
        boja = (0, 255, 0)     # ZELENA za ugodno
    else:
        boja = (255, 0, 0)     # CRVENA za vruće/toplo

    # Oboji svih 7 lampica na traci u odabranu boju
    for i in range(7):
        traka[i] = boja
    traka.show()

sleep(20) # Kratka pauza u petlji radi stabilnosti radija
```

Uređaj 2: PRIJAMNIK (Sada s ugrađenim zvukom i alarmom)

RGB traka je na pinu **P0**. Prilikom svakog ugodnog ili hladnog mjerenja čut će se kratki ugodni ton. Ako temperatura prijeđe 26°C, traka će zasvijetliti crveno, a ugrađeni zvučnik će odsvirati brzi alarm za uzbunu.

```
from microbit import *

import radio

import neopixel

import music


radio.on()

radio.config(channel=30)


traka = neopixel.NeoPixel(pin0, 7)


# Početno gašenje trake

for i in range(7): traka[i] = (0, 0, 0)

traka.show()


while True:

    poruka = radio.receive()


    if poruka:

        temp = int(poruka)

        display.scroll(str(temp) + "C")


        # --- ODREĐIVANJE BOJE I ZVUKA ZAVISNO O TEMPERATURI ---

        if temp < 22:
```

```
        boja = (0, 0, 255)     # PLAVA za hladno
        for i in range(7): traka[i] = boja
        traka.show()
        # Kratki niski ton za potvrdu normalnog mjerenja
        music.pitch(440, duration=100)

    elif temp >= 22 and temp <= 26:
        boja = (0, 255, 0)     # ZELENA za ugodno
        for i in range(7): traka[i] = boja
        traka.show()
        # Kratki srednji ton za potvrdu normalnog mjerenja
        music.pitch(554, duration=100)

    else:
        boja = (255, 0, 0)     # CRVENA za vruće (ALARM)
        for i in range(7): traka[i] = boja
        traka.show()
        # Sviraj ugrađenu brzu melodiju upozorenja jer je prevruće!
        music.play(music.BA_DING)

sleep(20)
```

Uređaj 2: PRIJAMNIK (Pametna klima s ventilatorom, RGB trakom i zvukom)

- **Spajanje modula na Boson ploču:**
    - **RGB traka** -> pin **P0**
    - **Ventilator modul** -> pin **P1**

```
from microbit import *
import radio
import neopixel
import music

radio.on()
radio.config(channel=30)

# Pripremi RGB traku na pinu P0
traka = neopixel.NeoPixel(pin0, 7)

# Na početku ugasi traku i ventilator
for i in range(7): traka[i] = (0, 0, 0)
traka.show()
pin1.write_digital(0) # Ventilator isključen (0)

while True:
    poruka = radio.receive()

    if poruka:
        temp = int(poruka)
```

```
display.scroll(str(temp) + "C")

# --- KONTROLA KLIMATIZACIJE ZAVISNO O TEMPERATURI ---
if temp < 22:
    # PLAVA BOJA - Ventilator ugašen
    for i in range(7): traka[i] = (0, 0, 255)
    traka.show()
    pin1.write_digital(0) # Ugasi ventilator
    music.pitch(440, duration=100)

elif temp >= 22 and temp <= 26:
    # ZELENA BOJA - Ventilator ugašen
    for i in range(7): traka[i] = (0, 255, 0)
    traka.show()
    pin1.write_digital(0) # Ugasi ventilator
    music.pitch(554, duration=100)

else:
    # CRVENA BOJA - VRUĆE JE!
    for i in range(7): traka[i] = (255, 0, 0)
    traka.show()

    # UPALI VENTILATOR (1) da hladi sobu
    pin1.write_digital(1)

    # Alarm upozorenja
```

```
    music.play(music.BA_DING)

sleep(20)
```

Bežični odašiljač smajlića (Radio slanje emocija) 😊 -> 😱

U ovoj vježbi koristiš dva micro:bita bez ikakvih vanjskih Boson modula. Stiskom na gumb A ili B šalješ smajlića kroz zrak, a drugi micro:bit ga odmah nacrta na svom ekranu.

Kôd za ODAŠILJAČ:

```
from microbit import *

import radio


radio.on()

radio.config(channel=40)


while True:

    if button_a.was_pressed():

        radio.send("sretan")

    if button_b.was_pressed():

        radio.send("tuzan")

    sleep(50)
```

Kôd za PRIJAMNIK:

```
from microbit import *

import radio


radio.on()

radio.config(channel=40)


while True:

    poruka = radio.receive()
```

```
if poruka == "sretan":
    display.show(Image.HAPPY)
elif poruka == "tuzan":
    display.show(Image.SAD)


sleep(20)
```

Za ovu vježbu potreban ti je **1 Odašiljač (Daljinski)** i **3 zasebna Prijamnika (Baze)**. Na svakom prijamniku je **RGB traka spojena na pin P0**.

Evo kako radi sustav:

- Kada na Odašiljaču stisneš gumb **A**, on šalje poruku "1". Prva baza se pali, ostale se gase.
- Kada stisneš gumb **B**, šalje poruku "2". Druga baza se pali, ostale se gase.
- Kada stisneš **A i B zajedno**, šalje poruku "3". Treća baza se pali, ostale se gase.

Uređaj 1: ODAŠILJAČ (Daljinski upravljač)

Ovaj kôd učitavaš na **jedan micro:bit** koji će služiti kao kontroler.

```
from microbit import *
import radio

radio.on()
radio.config(channel=50) # Svi uređaji moraju biti na kanalu 50

while True:
    # Ako stisneš oba gumba odjednom -> Šalje broj 3
    if button_a.is_pressed() and button_b.is_pressed():
        radio.send("3")
        display.show("3")
        sleep(500)
    # Ako stisneš samo gumb A -> Šalje broj 1
    elif button_a.was_pressed():
        radio.send("1")
        display.show("1")
    # Ako stisneš samo gumb B -> Šalje broj 2
```

```
elif button_b.was_pressed():
    radio.send("2")
    display.show("2")

sleep(50)
```

PRIJAMNICI (Baze s RGB trakama na pinu P0)

Za tri prijamnika koristit ćeš **gotovo isti kôd**. Jedina razlika je u varijabli MOJ_BROJ na samom početku. Kada otvoriš micro:bit Python Editor, za svaku pločicu promijeni taj broj prije učitavanja.

Kôd za PRIJAMNIK 1 (Upiši **MOJ_BROJ = "1"**)

```
from microbit import *

import radio

import neopixel


radio.on()

radio.config(channel=50)

traka = neopixel.NeoPixel(pin0, 7)


# --- OVDJE ODREĐUJEŠ KOJA JE OVO BAZA (Za prvu pločicu stavi "1") ---

MOJ_BROJ = "1"

display.show(MOJ_BROJ)


while True:

    poruka = radio.receive()


    if poruka:

        # Ako je primljeni broj jednak broju ove baze -> UPALI SVJETLO

        if poruka == MOJ_BROJ:

            for i in range(7): traka[i] = (255, 0, 0) # Crvena boja

            traka.show()

        # Ako je primljen neki drugi broj -> UGASI SVJETLO
```

```
    else:
        for i in range(7): traka[i] = (0, 0, 0)
        traka.show()


sleep(20)
```

Kôd za PRIJAMNIK 2 (Upiši **MOJ_BROJ = "2"**)

```
from microbit import *
import radio
import neopixel

radio.on()
radio.config(channel=50)
traka = neopixel.NeoPixel(pin0, 7)

# --- ZA DRUGU PLOČICU PROMIJENI NA "2" ---
MOJ_BROJ = "2"
display.show(MOJ_BROJ)

while True:
    poruka = radio.receive()

    if poruka:
        if poruka == MOJ_BROJ:
            for i in range(7): traka[i] = (0, 255, 0) # Zelena boja
            traka.show()
        else:
            for i in range(7): traka[i] = (0, 0, 0)
            traka.show()

    sleep(20)
```

Kôd za PRIJAMNIK 3 (Upiši **MOJ_BROJ = "3"**)

```
from microbit import *
import radio
import neopixel

radio.on()
radio.config(channel=50)
traka = neopixel.NeoPixel(pin0, 7)

# --- ZA TREĆU PLOČICU PROMIJENI NA "3" ---
MOJ_BROJ = "3"
display.show(MOJ_BROJ)

while True:
    poruka = radio.receive()

    if poruka:
        if poruka == MOJ_BROJ:
            for i in range(7): traka[i] = (0, 0, 255) # Plava boja
            traka.show()
        else:
            for i in range(7): traka[i] = (0, 0, 0)
            traka.show()

    sleep(20)
```

- Poredaj tri prijamnika s RGB trakama na stol i upali ih (na ekranima će im pisati 1, 2 i 3).
- Uzmi odašiljač i stisni gumb A. Vidjet ćeš kako se u istom trenutku pali **samo prva traka**, dok ostale miruju.
- Stisni gumb B – prva se gasi, a **pali se druga traka**.
- Stisni oba gumba odjednom – **pali se treća traka**, a prva i druga su ugašene.

Vježba: "Puhalica" za svjećice (Ugasiti svjetlo puhanjem)

Zamisli da je tvoja RGB traka rođendanska svijeća. Na početku sve lampice svijetle jarko žuto. Tvoj zadatak je snažno puhnuti izravno u mikrofon micro:bita kako bi "ugasio svjećice"!

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)


# Na početku upali traku u žutu boju (svjećica gori)

for i in range(7): traka[i] = (255, 150, 0)

traka.show()

display.show(Image.HAPPY)


while True:

    # Ako snažno puhneš u mikrofon, glasnoća skače blizu maksimuma

    if microphone.sound_level() > 180:

        display.show(Image.SURPRISED)


        # Ugasi lampice jednu po jednu (efekt puhanja)

        for i in range(7):

            traka[i] = (0, 0, 0)

            traka.show()

            sleep(100)


        display.show(Image.ASLEEP)

        sleep(3000) # Čekaj 3 sekunde prije nego što se svijeća opet upali
```

```
    # Ponovno upali svijeću
    for i in range(7): traka[i] = (255, 150, 0)
    traka.show()
    display.show(Image.HAPPY)

sleep(20)
```

Ekološki "Pametni prozor" (Mikrofon + Ventilator) 🌬️👏

- **Moduli:** RGB traka -> **P0**, Ventilator -> **P1**
- **Opis:** Simulacija pametnog prozora koji provjetrava sobu. Kada glasno pljesneš ili vikneš prema mikrofonu, ventilator se upali na 5 sekundi kako bi unio svjež zrak, a RGB traka svijetli plavo (simbol zraka). Nakon toga se sve automatski gasi.

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)

pin1.write_digital(0) # Ventilator ugašen


while True:

   # Ako mikrofon uoči glasan pljesak/zvuk

   if microphone.sound_level() > 110:

      display.show(Image.SURPRISED)


      # Upali ventilator i oboji traku u cijan/plavu boju

      pin1.write_digital(1)

      for i in range(7): traka[i] = (0, 255, 255)

      traka.show()


      sleep(5000) # Provjetravanje traje 5 sekundi


      # Ugasi sve natrag

      pin1.write_digital(0)

      for i in range(7): traka[i] = (0, 0, 0)
```

```
    traka.show()
    display.clear()

sleep(50)
```

Igra "Tiha ruka" (Test stabilnosti s mikrofonom) 🤫🕹️

- **Moduli:** RGB traka -> **P0**
- **Opis:** Na početku igre sve lampice na traci svijetle zeleno. Tvoj zadatak je uzeti micro:bit u ruku i pomicati ga po sobi ili stolu što tiše možeš. Ako napraviš bilo kakav šum (škripanje stolice, preglasan korak, kašalj), jedna po jedna lampica na traci postat će crvena. Igra završava kad sve lampice "izgore" u crveno!

```
from microbit import *

import neopixel


traka = neopixel.NeoPixel(pin0, 7)

pogreske = 0


# Na početku sve svijetli zeleno

for i in range(7): traka[i] = (0, 255, 0)

traka.show()


while True:

    if pogreske < 7:

        # Ako uoči buku veću od uobičajene dok se mičeš

        if microphone.sound_level() > 90:

            # Oboji trenutnu lampicu u crveno kao kaznu

            traka[pogreske] = (255, 0, 0)

            traka.show()

            pogreske += 1

            sleep(500) # Pauza da ne broji istu buku više puta

    else:
```

```
    # KRAJ IGRE - Izgubio si, cijela traka bljeska crveno
    display.show(Image.SAD)
    for _ in range(3):
        for i in range(7): traka[i] = (255, 0, 0)
        traka.show()
        sleep(200)
        for i in range(7): traka[i] = (0, 0, 0)
        traka.show()
        sleep(200)

    # Resetiraj igru
    pogreske = 0
    for i in range(7): traka[i] = (0, 255, 0)
    traka.show()
    display.clear()

sleep(50)
```

Bežični slavljenik (Radio + Mikrofon + RGB traka) 🎉🎤

- **Opis:** Ako imaš dva micro:bita, ova vježba je super za zabavu. Jedan micro:bit je na stolu pokraj tebe i on sluša tvoj glas ili pljesak kroz mikrofon. Čim glasno vikneš ili zaplješćeš, on bežično kroz zrak pošalje signal drugom micro:bitu (koji je spojen na Boson ploču), a taj drugi micro:bit napravi pravi svjetlosni vatromet na svojoj RGB traci!

Kôd za UREĐAJ 1 (Slušač):

```
from microbit import *
import radio

radio.on()
radio.config(channel=60)

while True:
    # Ako vikneš ili pljesneš blizu mikrofona
    if microphone.sound_level() > 120:
        radio.send("vatromet")
        display.show("*")
        sleep(1000) # Pauza prije idućeg slanja
        display.clear()
    sleep(50)
```

Kôd za UREĐAJ 2 (Vatromet s RGB trakom na P0):

```
from microbit import *
import radio
import neopixel
import random
```

```
radio.on()
radio.config(channel=60)
traka = neopixel.NeoPixel(pin0, 7)

while True:
    poruka = radio.receive()

    if poruka == "vatromet":
        # Napravi 10 brzih nasumičnih bljeskova u boji
        for _ in range(10):
            r = random.randint(0, 255)
            g = random.randint(0, 255)
            b = random.randint(0, 255)

            for i in range(7):
                traka[i] = (r, g, b)
            traka.show()
            sleep(100)

        # Ugasi nakon vatrometa
        for i in range(7): traka[i] = (0, 0, 0)
        traka.show()

    sleep(20)
```