

Sada úloh pro práci s LED páskem připojeným k micro:bitu

Sada úloh je vytvořena pro žáky na druhém stupni základní školy. Časová dotace by měla být pět vyučovacích hodin, ale záleží na tom jak jsou žáci zdatní v programování a práci s micro:bitem. Žáci by měli znát blokové programování typu Scratch a v něm by měli znát hlavně práci s proměnnými, cykli a podmínkami. Žáci by také měli mít několik předchozích hodin věnovaných práci se samotným micro:bitem a prostředím MakeCode, aby se lépe orientovali v tomto prostředí a zvládli samostatně pracovat na úlohách.

Příprava před začátkem úloh: Připojte LED pásek do portu **P0** na micro:bitu. Toto zapojení budeme používat ve všech úlohách. Do prostředí MakeCode musíte přidat rozšíření, abychom mohli s LED páskem pracovat. Rozšíření se jmenuje **Neopixel**. Z tohoto rozšíření budeme vždy používat blok pro počáteční nastavení pásku, který nám nastaví proměnnou „strip“ na LED pásek o určitém počtu LED diod.



Část první – Začínáme s LED páskem

Úloha 1 – Rozsvěčíme pásek

Zadání úlohy: Rozsviďte celý pásek určitou barvou z předpřipraveného seznamu barev, které jsou v bloku „show color“.

Rozšiřující zadání: Rozsviďte pásek svojí vlastní barvou, kterou namícháte změněním jednotlivých RGB hodnot.

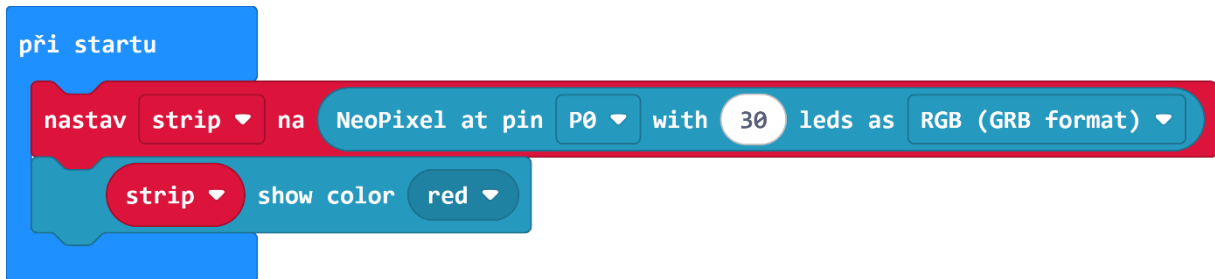
Informace k rozšířenému zadání úlohy:

- Blok pro změnění RGB hodnot je v bloku Neopixel pod záložkou více.
- Míchání vlastních barev si můžete vyzkoušet na této webové stránce: <https://www.csfieldguide.org.nz/en/interactives/rgb-mixer/>

Cíle úlohy: Žák zvládne rozsvítit celý pásek jednou z předurčených barev a také svojí vlastní barvou v RGB formátu.

Metodické poznámky: Základem této úlohy je, aby byli žáci schopni připojit LED pásek k micro:bitu a zobrazovat na něm určité barvy. Jak ty, které jsou předpřipravené v rozšíření pro Neopixel, tak i namícháním vlastních barev pomocí RGB hodnot. Žáci si zde mohou jednoduše vyzkoušet, jak se budou barvy měnit při změně RGB hodnot. Učitel se může následně ptát žáků jaké hodnoty bude mít například světle modrá barva: rgb(173,216,230), růžová barva: rgb(255,192,203), žlutá barva: rgb(255,255,0), fialová barva: rgb(255,0,255).

Vzorové řešení:



Úloha 2 – Střídající se barvy

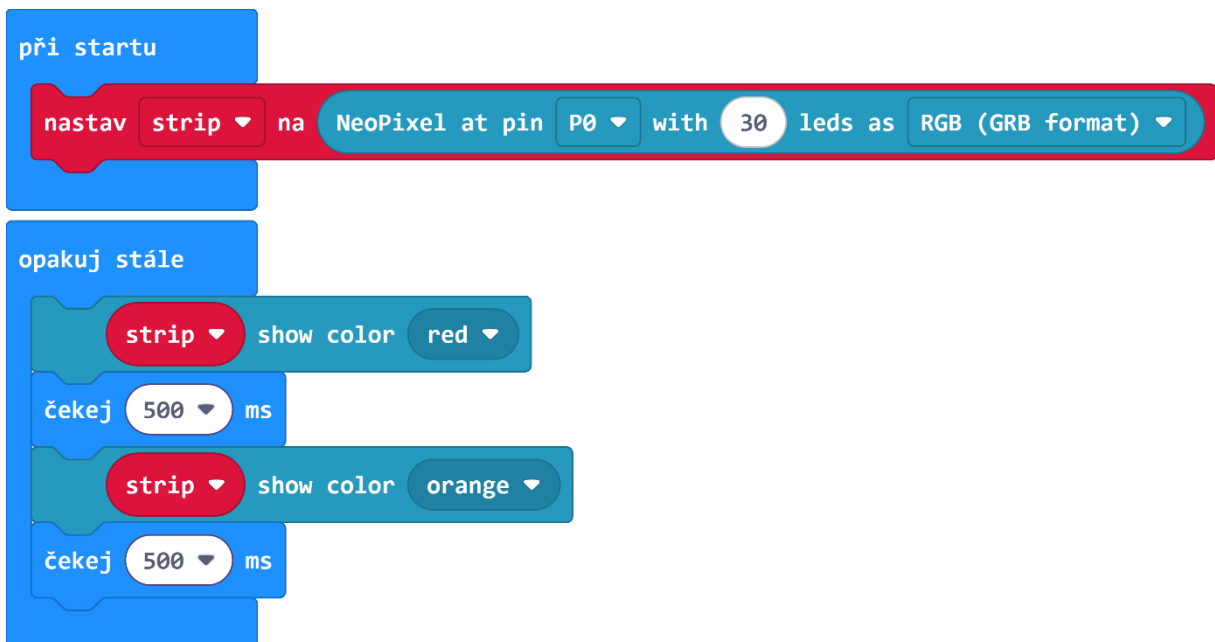
Zadání úlohy: Vytvořte program, který bude v celé délce pásku měnit dvě různé barvy po určitém časovém intervalu.

Rozšiřující zadání: Upravte program, aby střídal barvy duhy ve správném pořadí za sebou po určitém časovém intervalu v celé délce pásku.

Cíle úlohy: Žák je schopen měnit barvy na LED pásku po určitém časovém intervalu.

Metodické poznámky: Žáci si mohou jako rozšíření úlohy vyzkoušet měnit barvy jenom několikrát a nenechat pásek měnit barvy neustále.

Vzorové řešení:



Část druhá – Práce s pozicemi na LED pásku

Úloha 3 – Časovač

Zadání úlohy: Sestavte program, který umožní pásku fungovat jako časovač. Pracovat bude tak, že se každou vteřinu vypne jedna LED dioda, dokud nebude pásek celý zhasnutý a poté by se měl znovu celý rozsvítit a časovač bude pokračovat.

Rozšiřující zadání: Upravte předchozí program, tak aby fungoval jako časovač pro čištění zubů. Stlačením tlačítka micro:bitu se časovač spustí a měl by běžet celkově dvě minuty.

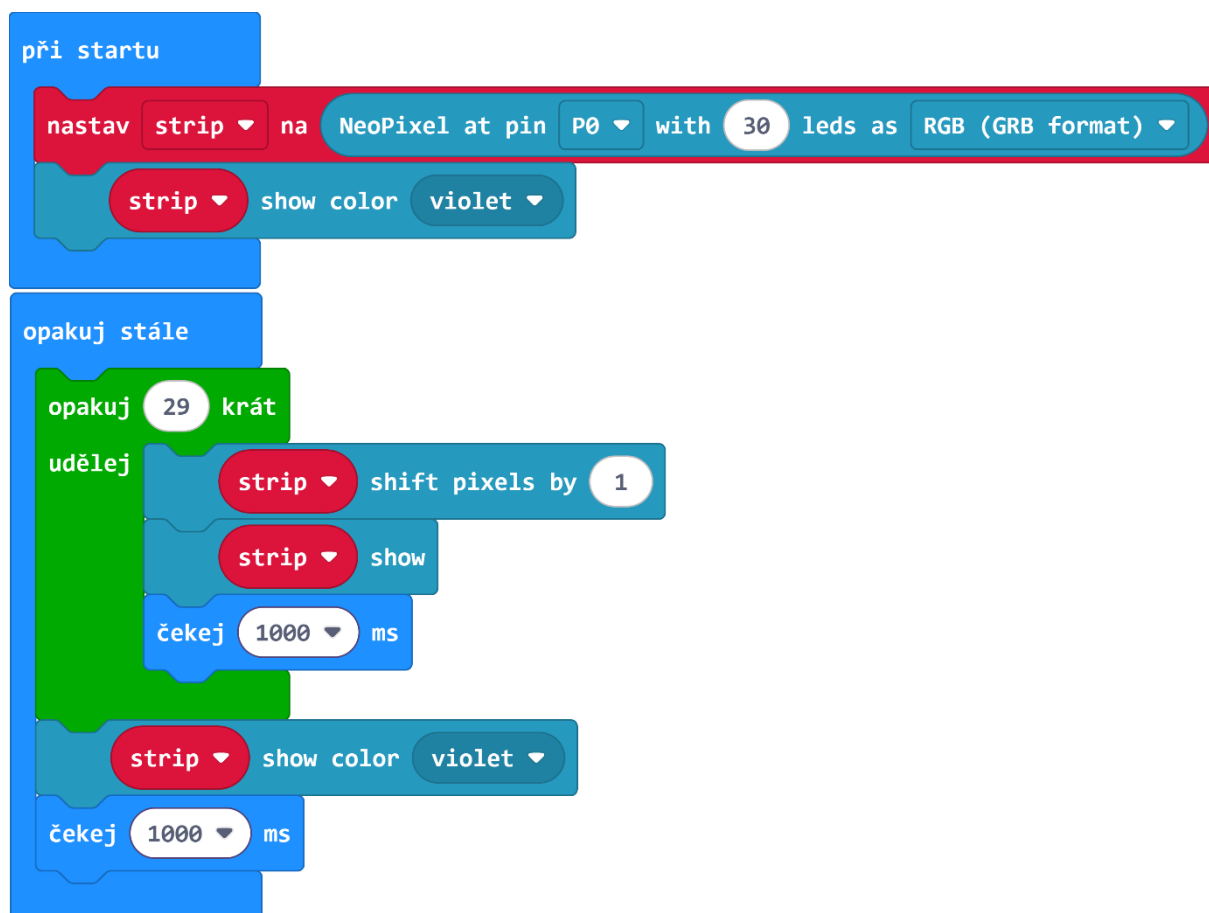
Informace k zadání úlohy:

- Zhasínání diody se provádí přes blok „shift pixels“.
- Každá změna na pásku se také musí určitým blokem zobrazovat.

Cíle úlohy: Žák zvládne vytvořit funkční časovač za pomoci LED pásku.

Metodické poznámky: Hlavní problémem pro žáky může být, že v opakování nebudou zobrazovat změny na pásku pomocí bloku „strip show“. Nestačí pouze vypínat pixely přes blok „shift pixels“.

Vzorové řešení:

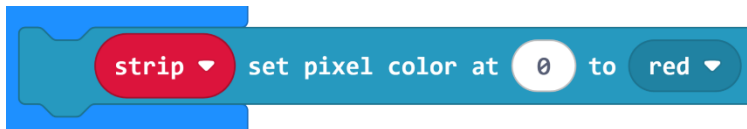


Úloha 4 – Střídáme dvě barvy

Zadání úlohy: Vytvořte program, který bude střídat dvě barvy po určitém počtu diod ve stejně velkých intervalech.

Informace k zadání úlohy:

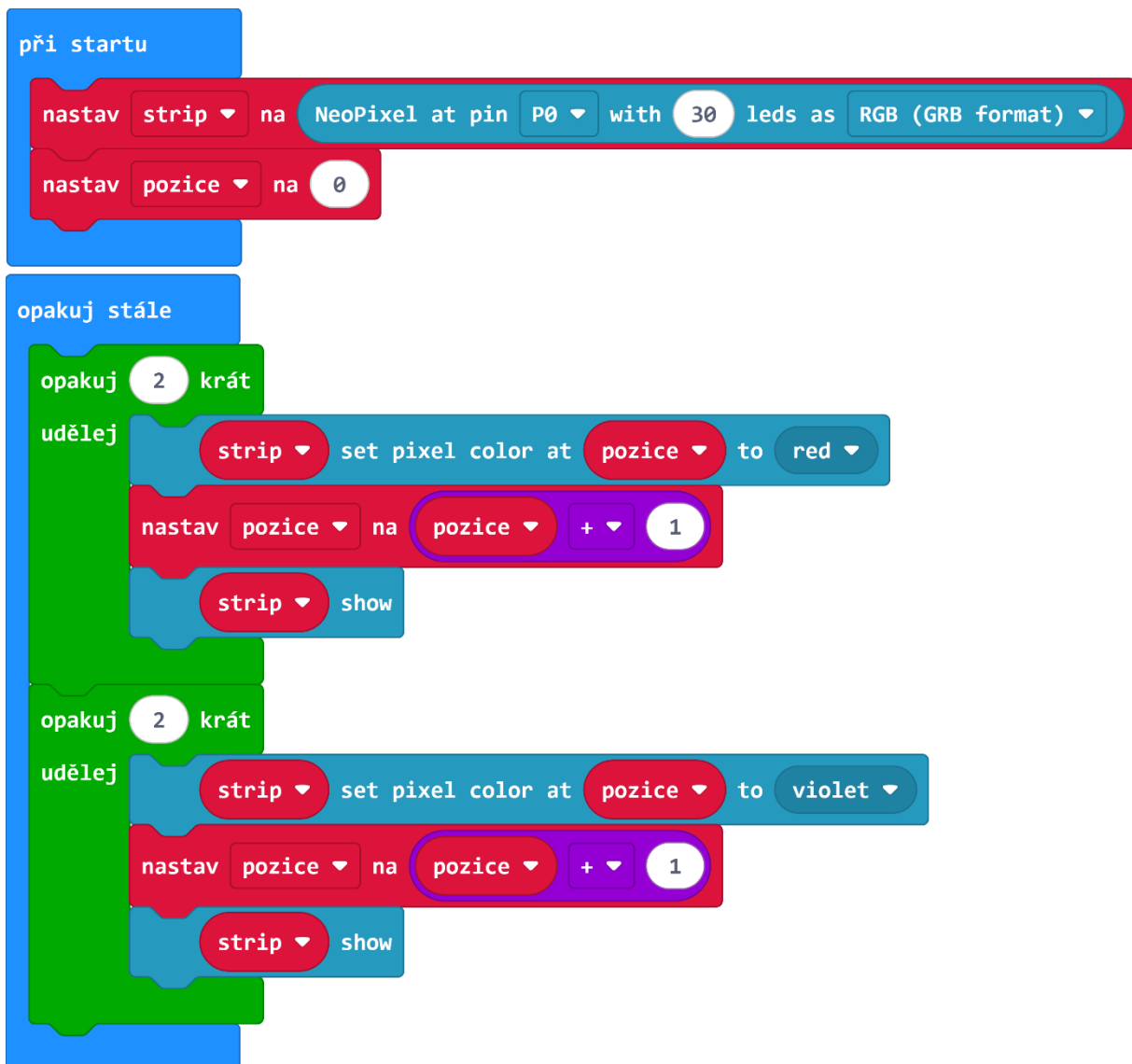
- V programu musíte k jednotlivým diodám přistupovat pomocí pozice, která je na pásku číslovaná od nuly a nastavit jim příslušnou barvu přes tento blok:



Cíle úlohy: Žák je schopen vytvořit program, v kterém se barvy střídají ve stejně velkých intervalech.

Metodické poznámky: Žáci si mohou vyzkoušet střídání barev po zvětšujících se intervalech pomocí měnění počtu opakování u jednotlivých barev. Žák nesmí zapomenout, že v jednom kroku musí nastavit barvu pro určitý pixel, zobrazit jí a následně i zvětšit index pro zobrazení barvy v dalším kroku. Některý žák může omezit hodnoty proměnné, aby se nezvětšovaly poté co proměnná nabyde čísla 30. A toto řešení můžete ukázat celé třídě jako nejlepší možné, ale pokud žáci neomezí zvětšování proměnné, není to problém, protože micro:bit zvládne pracovat s 32-bitovými čísly a k takovému číslu se žáci v hodině nedostanou ani když hodnoty neomezí.

Vzorové řešení:



Úloha 5 – Každá třetí červená

Zadání úlohy: Vytvořte program, který rozsvítí pásek tak, aby každá třetí dioda svítila červeně a ostatní diody zeleně. Program by měl fungovat pro libovolně dlouhý pásek.

Rozšiřující zadání: Zjistěte, jaké jsou možné další kombinace rozsvěcování pásku v pravidelných intervalech, tak aby zelená barva začínala na pásku a zároveň červená končila na poslední pozici na pásku.

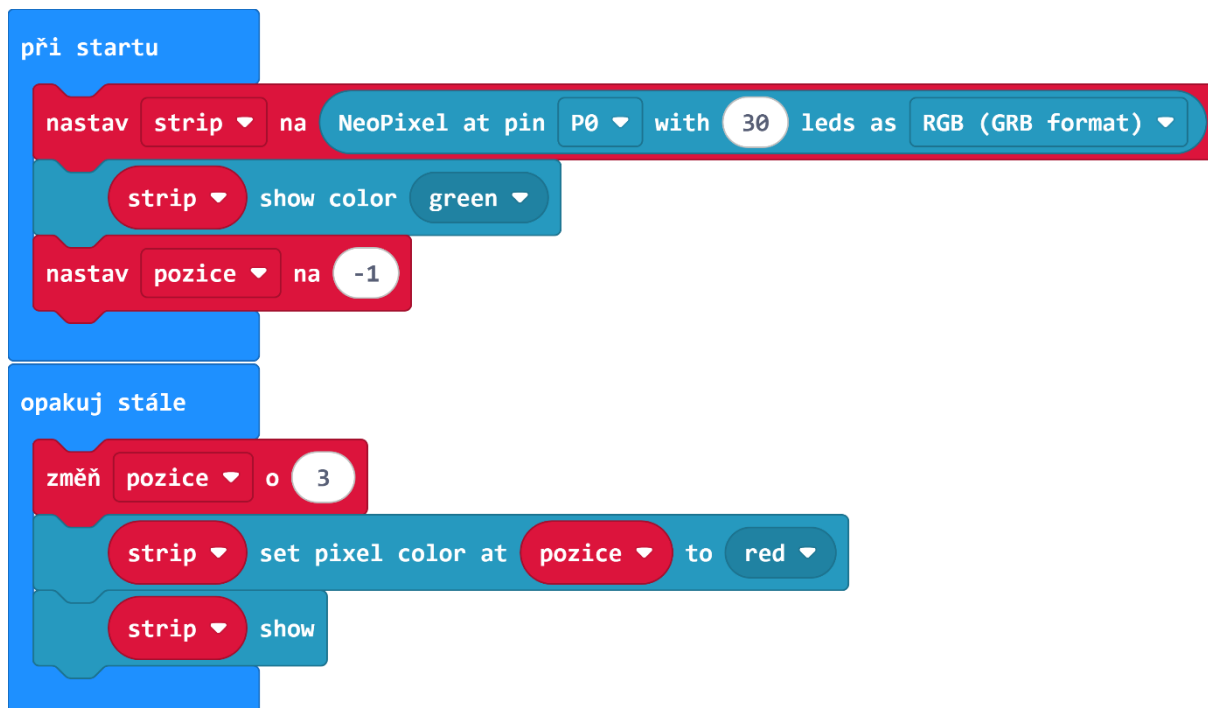
Informace k zadání úlohy:

- Při sestavování programu nepoužívejte cyklů ze sekce smyček.
- Zelená barva by měla na pásku začínat a červená barva by měla pásek na konci uzavírat.

Cíl úlohy: Žák je schopen střídat barvy po určitém počtu diod, bez použití cyklů.

Metodické poznámky: Žáci musejí na začátku rozsvítit celý pásek jednou barvou a poté na určitém indexu rozsvítit diodu jiné barvy a také index postupně zvětšovat. Aby se barvy střídaly v pravidelných intervalech je potřeba na začátku hodnotu proměnné „pozice“ nastavit na -1.

Vzorové řešení:



Úloha 6 – Pohybující se duha

Zadání úlohy: Vytvořte program, který bude na pásku zobrazovat neustále se pohybující se duhu.

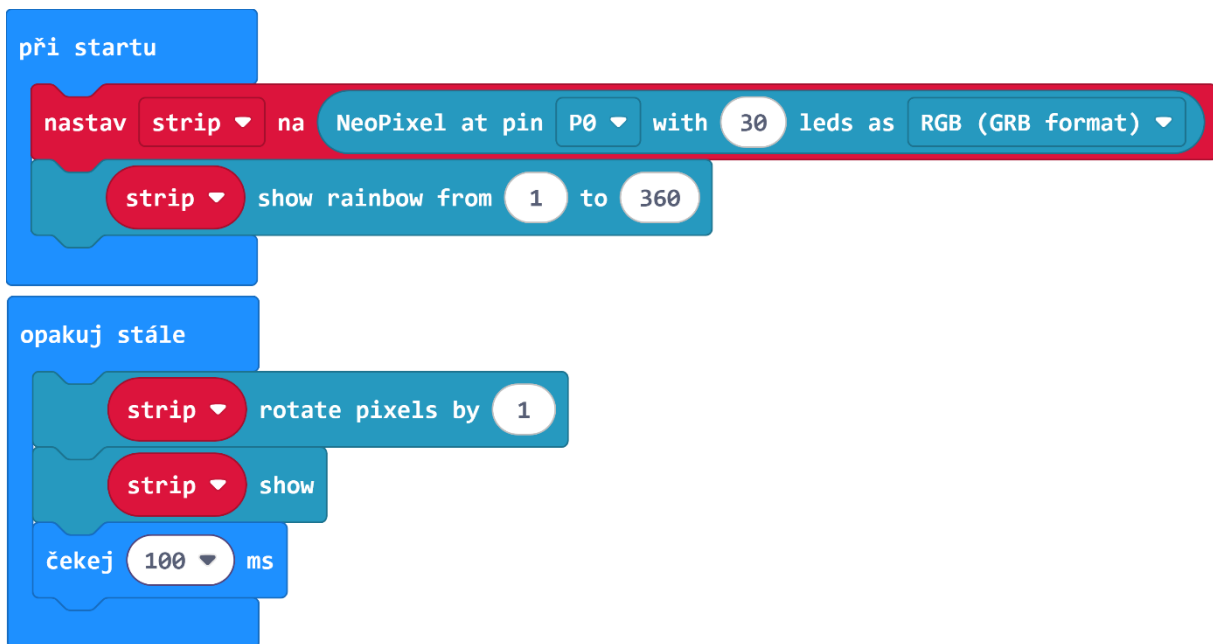
Informace k zadání úlohy:

- Pro zobrazení duhy je připravený blok „show rainbow“. Tento blok umístěte do bloku „při startu“.
- Pohybování duhy můžete docílit blokem z rozšíření Neopixel, který slouží pro rotaci pixelů.
- Do bloku „opakuji stále“ také umístěte blok „čkej“, aby byly vidět změny barvy na jednotlivých diodách.

Cíle úlohy: Žák je schopen rozpohybovat duhu na pásku použitím správného bloku i když s ním předtím nepracoval.

Metodické poznámky: Pro co nejhladší pohyb je lepší dát posun pixelů pouze o jednu diodu a do kódu přidat blok „čkej“. Žáci si také mohou vyzkoušet zadat do bloku „rotate pixels“ záporné číslo, aby zjistili, že se duha bude pohybovat na opačnou stranu než doposud.

Vzorové řešení:

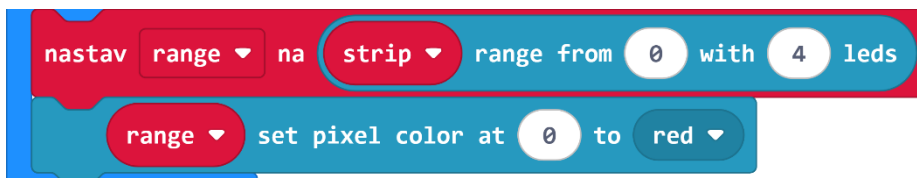


Úloha 7 – Půlení pásku

Zadání úlohy: Vytvořte program, ve kterém se budou postupně rozsvěcovat a zhasínat diody od každého konce pásku, než se potkají uprostřed.

Informace k zadání úlohy:

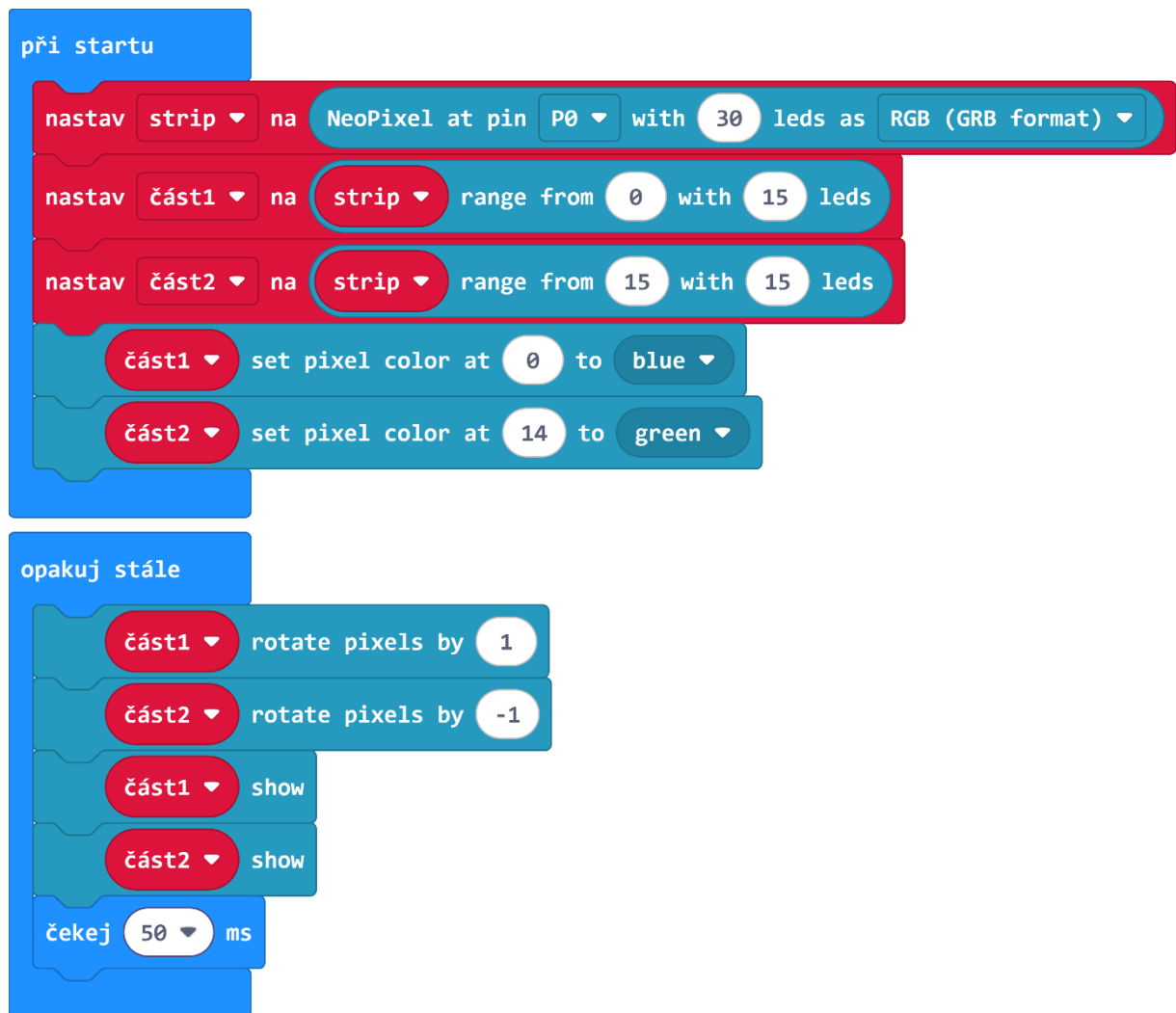
- K rozpůlení pásku využijte připraveného bloku v rozšíření Neopixel.
- První blok rozdělí pásek na část, která začíná na pozici nula a bude obsahovat čtyři LED diody.
- Druhý blok rozsvítí první diodu ve vytvořené části červeně.



Cíle úlohy: Žák je schopen rozdělit pásek do dvou stejně velkých částí a využít stejný princip posunu rozsvícení diod jako v úloze číslo 6.

Metodické poznámky: Žákům může dělat problém nastavování částí pásku, když první začíná na pozici 0 s 15 diodami a druhé začíná na pozici 15 s 15 diodami. Dále musejí nastavit barvu prvního pixelu z první části a posledního pixelu z druhé části. A tyto pixely mají v jednotlivých částích pozice 0 a 14, což by mohly být jiné pozice, než očekávají, ale obě části jsou stejně velké s rozsahem prvků 0 až 14, jenom začínají na jiné pozici na pásku.

Vzorové řešení:



The image shows a Scratch script for controlling a NeoPixel LED strip. It is divided into two main sections: 'při startu' (when started) and 'opakuji stále' (repeat loop).

při startu

- nastav strip** na NeoPixel at pin P0 with 30 leds as RGB (GRB format)
- nastav část1** na strip range from 0 with 15 leds
- nastav část2** na strip range from 15 with 15 leds
- část1** set pixel color at 0 to blue
- část2** set pixel color at 14 to green

opakuji stále

- část1** rotate pixels by 1
- část2** rotate pixels by -1
- část1** show
- část2** show
- čekej** 50 ms

Část třetí – Ovládání LED pásku periferiemi

Úloha 8 – Ovládání toku duhy senzorem gest

Zadání úlohy: Vytvořte program, který bude pomocí senzoru gest ovládat tok duhy LED páskem. Při gestu vlevo se bude duha pohybovat z konce pásku směrem k začátku. Při gestu vpravo bude tento pohyb obrácený.

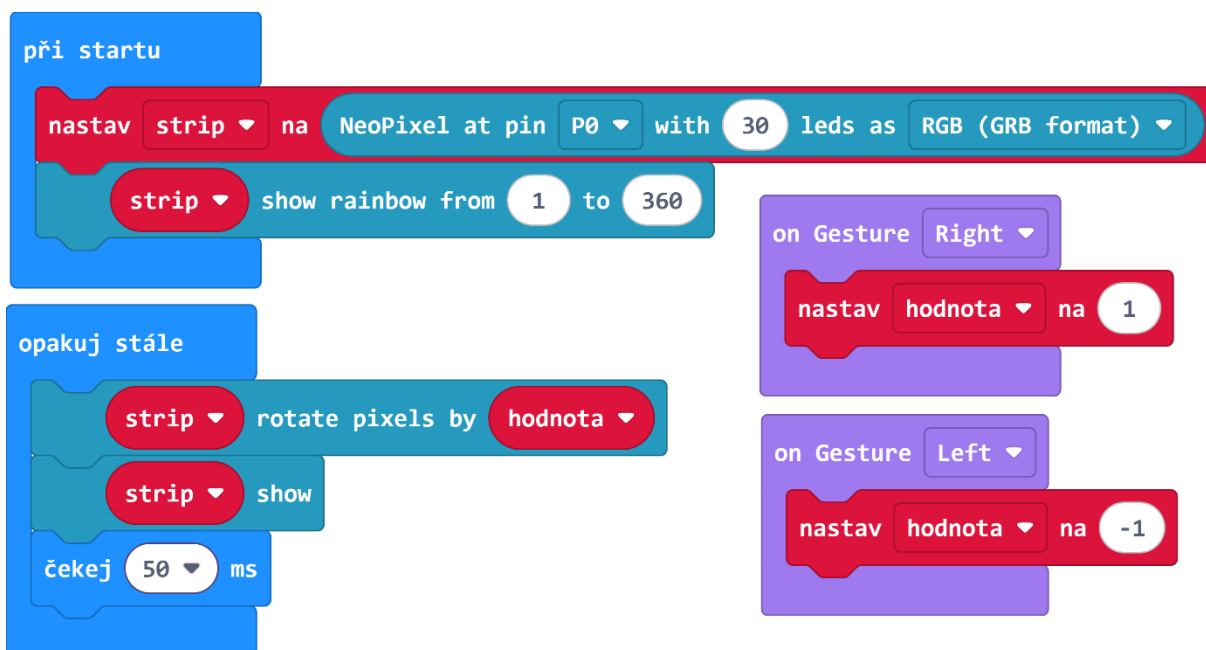
Informace k zadání úlohy:

- K micro:bitu připojte do portu **I2C** Gest senzor.
- Do prostředí MakeCode přidejte rozšíření **Grove**. V něm se nachází bloky pro práci se senzorem Gest.

Cíle úlohy: Žák je schopen využít senzor gest k ovládání pohybu duhy ve dvou různých směrech.

Metodické poznámky: Žáci si musí dát pozor na připojení senzoru gest do správného konektoru, protože se samotný senzor už nijak v programu nenastavuje. Pokud žáci nebudou chtít pracovat s duhou, mohou pohybovat i jednou rozsvěcující se diodou určité barvy.

Vzorové řešení:



Úloha 9A – Přípravná úloha k úloze 9 – funkce „mapuj“

Zadání úlohy: Vytvořte program, který bude pomocí funkce „mapuj“ zobrazovat hodnoty z rotačního snímače pohybu na displeji micro:bitu v rozsahu čísel 0 až 9.

Informace k zadání úlohy:

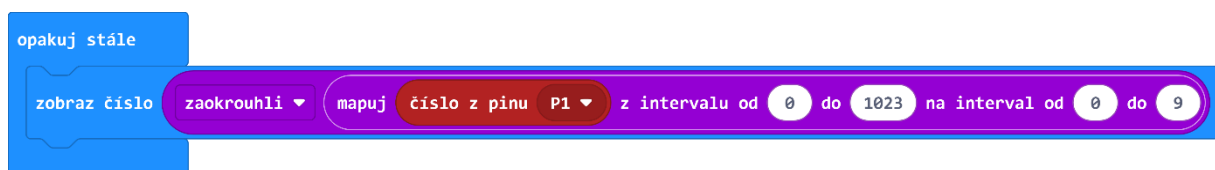
- K micro:bitu připojte rotační snímač úhlu do portu **P1**.
- Hodnotu pro přečtení hodnoty z portu **P1** najdete v záložce „Pokročilé“ v bloku „Piny“. Číslo z tohoto pinu si uložte do proměnné.

- Funkci mapuj najdete v bloku matematika. Ta funguje tak, že přenesse hodnoty z jednoho intervalu na hodnoty druhého intervalu.
- Prvním intervalem budou hodnoty senzoru, které jsou v rozpětí 0 až 1023. Druhým intervalem budou hodnoty zobrazované na displeji micro:bitu.
- Hodnoty zobrazované na micro:bitu upravte aby se zobrazovala pouze celá čísla.

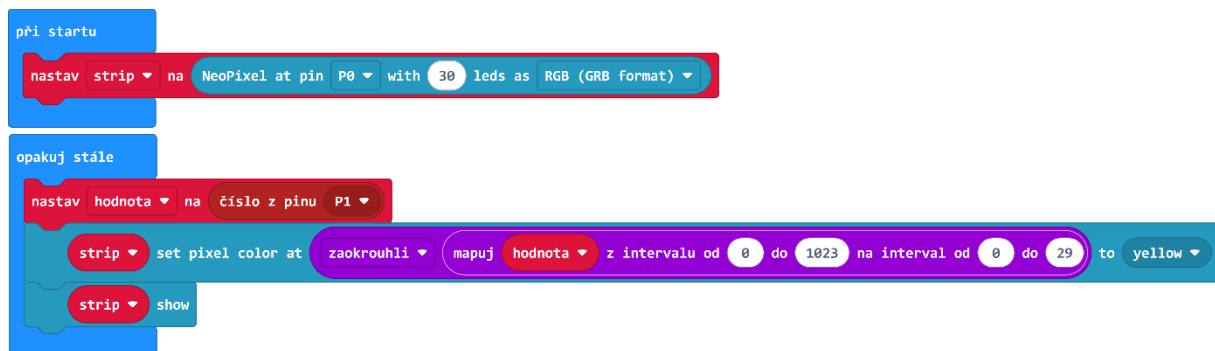
Cíle úlohy: Žák se naučí využívat funkci „mapuj“, kterou bude potřebovat v následující úloze.

Metodické poznámky: Pro zobrazení celých čísel se mohou žáci snažit zaokrouhlovat pouze hodnoty z rotačního snímače pohybu, protože se sním nesečkali a mohou se domnívat, že dodává i desetinné hodnoty. Desetinné hodnoty má ale za následek přímo funkce mapuj, a proto žáci musí zaokrouhlovat celou tuto funkci.

Vzorové řešení:



Úloha 9 – Plnění pásku otočným knoflíkem



Zadání úlohy: Na obrázku výše je program, který postupně plní LED pásek postupně se rozsvěčujícími se diodami na základě hodnot z otočného knoflíku. Tento program ale není úplně funkční. Zkuste před vytvářením tohoto programu zjistit proč program nefunguje optimálně a následně ho opravte.

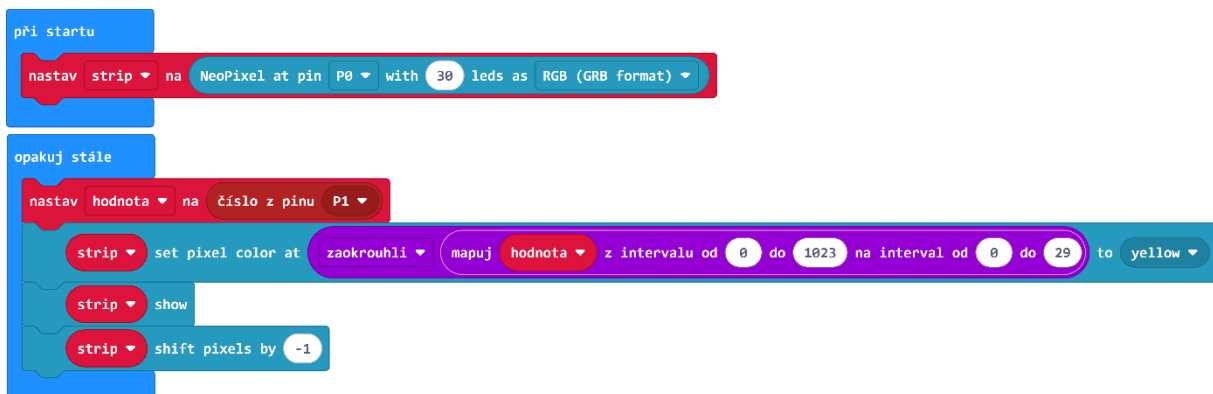
Informace k zadání úlohy:

- Druhým intervalem ve funkci „mapuj“ jsou v této úloze pozice první a poslední diody na LED pásku.

Cíle úlohy: Žák je schopen porozumět programu a jeho funkčnosti a základě toho program upravit, aby byl plně funkční.

Metodické poznámky: Funkce mapuj je zde zaokrouhlena, kvůli tomu, že žákům by jinak mohla proklívat nebo vůbec nesvítit poslední dioda na LED pásku. Dále se může stát, že žákům nebude při minimální hodnotě senzoru svítit první dioda. Pro vyřešení toho problému stačí umístit blok „shift pixels“ na konec v bloku „opakuji stále“. Jestliže žáci dají do bloku „shift pixels“ kladné číslo bude se jim pásek plnit od konce na začátek. Tento jev stačí opravit zadáním záporného čísla do tohoto bloku.

Vzorové řešení:



```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)

opakuje stále
nastav hodnota na číslo z pinu P1
strip set pixel color at zaokrouhli mapuj hodnota z intervalu od 0 do 1023 na interval od 0 do 29 to yellow
strip show
strip shift pixels by -1
```

Úloha 10 – Posouvání diody na základě hodnot snímače

Zadání úlohy: Upravte kód z úlohy devět, tak aby se po pásku pohybovala pouze jedna rozsvícená dioda.

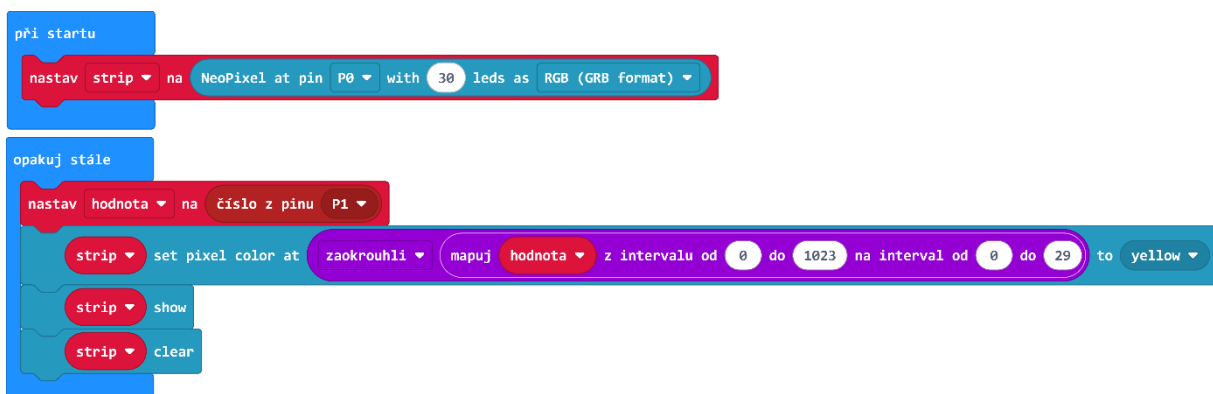
Informace k zadání úlohy:

- Aby se na pásku mohla pohybovat pouze jedna rozsvícená dioda budete potřebovat využít nový blok z rozšíření Neopixel, který zhasne všechny LED diody a tím zůstane svítit pouze dioda, která odpovídá aktuálnímu natočení knoflíku.

Cíle úlohy: Žák je schopen upravit kód z předchozí úlohy, použitím bloku „clear“ z rozšíření Neopixel, s kterým ještě nepracoval.

Metodické poznámky: Jako první se může stát, že žáci odstraní z kódu pouze blok „shift pixels“, čímž se bude rozsvěcovat pouze jedna dioda, ale také se nebude žádná zhasínat. Do kódu stačí místo bloku „shift pixels“ blok „clear“ z rozšíření Neopixel. Za blokem „clear“ není znovu vložen blok „show“, protože v tomto případě dochází ke stálému problikávání LED diody.

Vzorové řešení:



```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)

opakuje stále
nastav hodnota na číslo z pinu P1
strip set pixel color at zaokrouhli mapuj hodnota z intervalu od 0 do 1023 na interval od 0 do 29 to yellow
strip show
strip clear
```

Úloha 11 – Ovládání úrovně světelnosti na základě hodnot snímače

Zadání úlohy: Vytvořte program, který bude ovládat svítivost pásku na základě hodnot z otočného knoflíku, který snímá úhel.

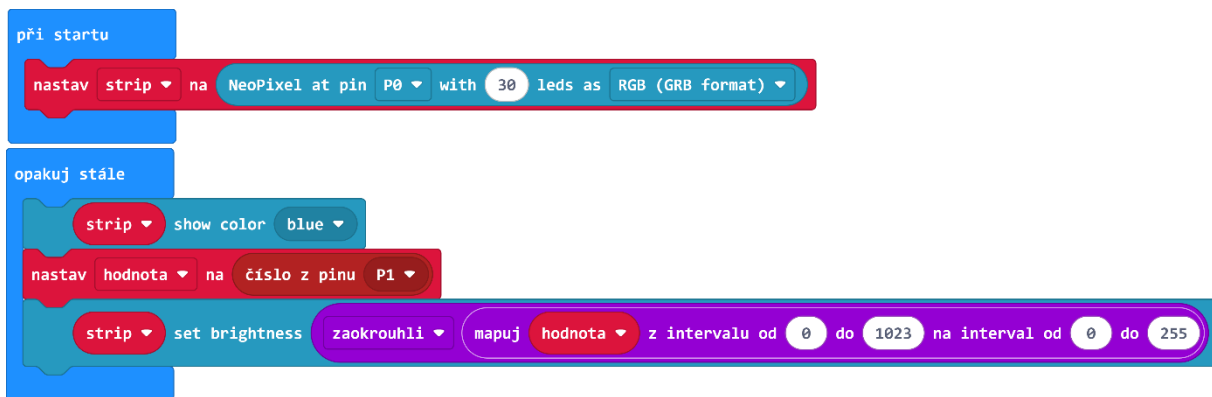
Informace k zadání úlohy:

- Svítivost pásku se může nastavovat v hodnotách 0 až 255.
- V této úloze musíte umístit blok pro rozsvícení pásku určitou barvou do bloku „opakuj stále“.

Cíle úlohy: Žák je schopen vytvořit program, který bude upravovat svítivost pásku na základě hodnot snímače.

Metodické poznámky: Problém při vytváření kódů může nastat v tom, že žák nastaví barvu LED pásku v bloku „při startu“ a tím mu nebude fungovat ovládání svítivosti. Nastavení barvy musí být v bloku „opakuj stále“, aby fungovalo ovládání svítivosti.

Vzorové řešení:



Úloha 12 – Ovládání světelnosti světelným senzorem

Zadání úlohy: Vytvořte program, který bude upravovat svítivost pásku na základě hodnot světelného senzoru.

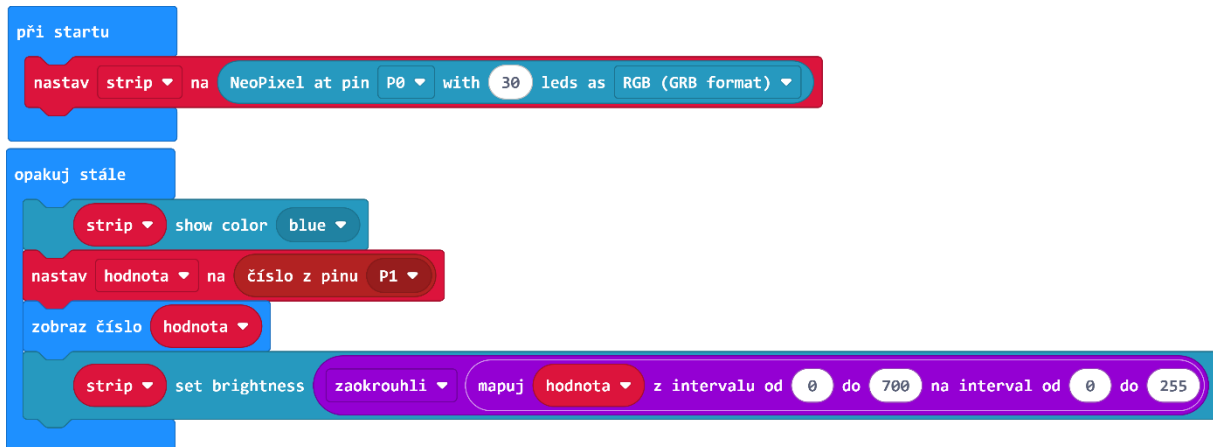
Informace k zadání úlohy:

- Připojte k micro:bitu světelný senzor do portu **P1**.
- Hodnoty ze senzoru si uložte do proměnné a zjistěte si maximální a minimální hodnoty, kterých dosahuje světelný senzor ve vašem prostředí.
- Minimální hodnota by měla být, když senzor přikryjete rukou a maximální, když na něj posvítíte baterkou nebo svítilnou telefonem. Tyto hodnoty si vypisujte na displej micro:bitu.
- Až budete mít rozsah hodnot světelného senzoru, tak je doplňte do funkce mapuj pro ovládání svítivosti pásku.

Cíle úlohy: Žák je schopen zjistit si hodnoty, kterých dosahuje senzor v jeho prostředí a využít těchto hodnot při tvorbě kódu.

Metodické poznámky: Pokud žáci nemohou ve škole používat mobilní telefon, bylo by vhodné pro tuto úlohu přinést do třídy svítilnu, aby měli hodnoty světelného senzoru větší rozsah a změny byly více vidět na svítivosti LED pásku.

Vzorové řešení:



Úloha 13 – Metr s ultrazvukovým senzorem

Zadání úlohy: Vytvořte program, který bude přenášet naměřenou vzdálenost ultrazvukovým senzorem na počet rozsvícených LED diod pásku. Poměr, v jakém se budou hodnoty přenášet si zvolte samy, ale pro začátek může každý centimetr představovat jednu rozsvícenou LED diodu.

Informace k zadání úlohy:

- K micro:bitu připojte ultrazvukový senzor do portu **P1** a hodnoty ze senzoru si uložte do proměnné.
- Pro práci s ultrazvukovým senzorem budeme potřebovat přidat rozšíření **Grove** do prostředí MakeCode.
- Hodnoty, které měří senzor si pro kontrolu můžete zobrazovat na displeji micro:bitu.
- Na pásku by měly být rozsvícené všechny diody až po aktuální hodnotu ze senzoru.

Cíle úlohy: Žák je schopen vytvořit realistický metr za pomoci LED pásku a ultrazvukového senzoru.

Metodické poznámky: Žákům může dělat problém, že nastavování barvy se zde musí dělat v bloku „opakuj stále“, a to pro pole diod, které se mají rozsvítit. A také nesmějí zapomenout přidat blok „clear“, aby se diody při každé změně vzdálenost mohly také zhasínat.

Vzorové řešení:

```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)

opakuji stále
nastav hodnota na Ultrasonic Sensor (in cm) at P1
nastav pole na strip range from 0 with mapuj zaokrouhli hodnota z intervalu od 0 do 30 na interval od 1 do 30 leds
pole show color purple
pole clear
```

Úloha 14 – Digitální metr

Zadání úlohy: Vytvořte program, který bude za pomoci připojeného displeje zobrazovat hodnoty ultrazvukového senzoru a tyto hodnoty bude přenášet na LED pásek.

Informace k zadání úlohy:

- Rozšiřte úlohu 13 tím, že k micro:bitu připojíte 4-digit displej do portu **P2**.
- Na displeji omezte zobrazované hodnoty pouze na ty, které umí zobrazovat váš LED pásek.
- Pro omezení hodnot můžete využít bloku v sekci matematika.

Cíle úlohy: Žák zvládne upravit předchozí úlohu, aby nově pracovala i s 4-digit displejem a také omezí hodnoty, které se na něm můžou zobrazovat.

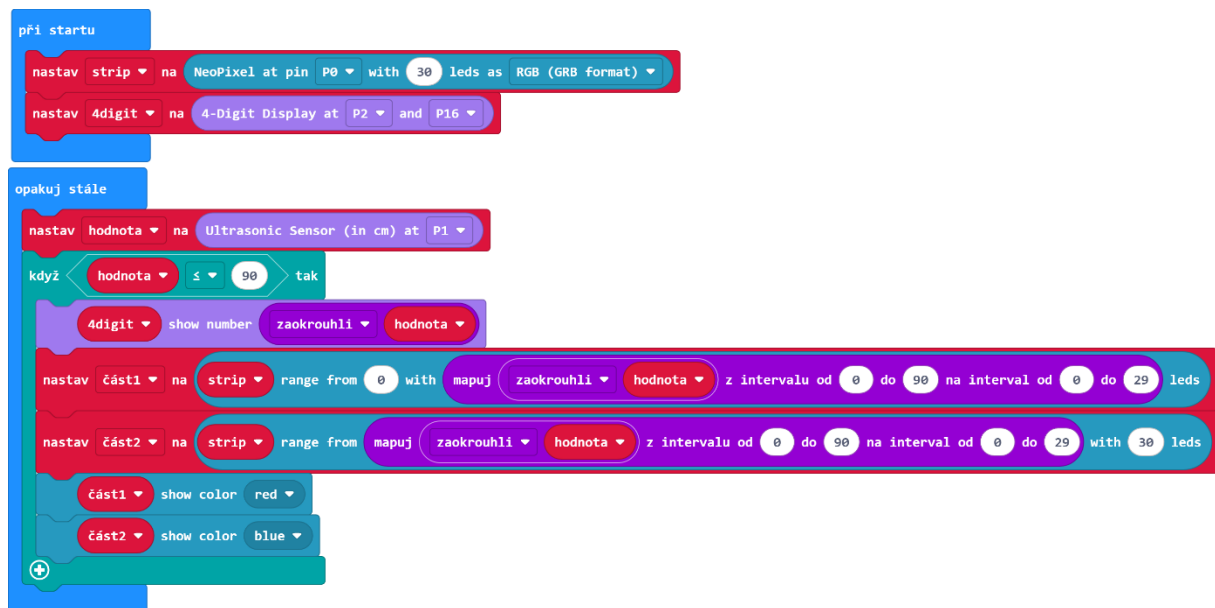
Metodické poznámky: Žáci mohou použít blok omez hodnotu z bloku matematika, s kterým se ještě nemuseli setkat. Jestli zvládnou omezit hodnoty zobrazované na displeji i jiným způsobem, než použitím bloku omez hodnotu je toto řešení také v pořádku. Pokud se žákům nebudou přesně rozsvěcovat diody, může být problém s hodnotami, protože ultrazvukový senzor vrací desetinné číslo. Je tedy potřeba hodnoty zaokrouhlovat.

Vzorové řešení:

```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)
nastav 4digit na 4-Digit Display at P2 and P16

opakuji stále
nastav hodnota na Ultrasonic Sensor (in cm) at P1
4digit show number zaokrouhli omez hodnotu hodnota na interval od 0 do 30
nastav pole na strip range from 0 with mapuj zaokrouhli hodnota z intervalu od 0 do 30 na interval od 1 do 30 leds
pole show color indigo
pole clear
```

Úloha 15 – Rozpoznání kódu



Zadání úlohy: Na základě kódu, který vidíte výše popište, co daný program dělá. Poté si kód vyzkoušejte a ověřte si své předchozí hypotézy.

Rozšiřující zadání: Tento kód má ovšem drobný nedostatek, který se může projevit při spuštění programu na micro:bitu. Pokud tento nedostatek odhalíte upravte kód, tak aby fungoval ve všech případech.

Cíle úlohy: Žák je schopen z jemu předloženému kódu, zjistit co program dělá, a ověřit si svá tvrzení.

Metodické poznámky: S vytvářením polí se žáci setkali už v úloze číslo sedm, pokud si nebudou jisti, jak pole fungují můžou si to znovu vyzkoušet v této úloze. Tato úloha navazuje na úlohu číslo 14, ale v prvním intervalu je zde rozsah od nuly do 90 centimetrů, aby žáci neměli dva stejné intervaly vedle sebe a lépe se v kódu orientovali. Pokud si žáci nebudou vědět rady s rozšiřujícím zadáním, můžete je navést otázkou co se děje s páskem, když není podmínka splněna. Žáci by měli odpovědět, že pásek zobrazí poslední hodnotu, pro kterou ještě podmínka platila. A pokud pořád nebudou vědět jaká chyba může nastat při spuštění programu, zeptejte se jich, co se s páskem stane, pokud podmínka nebude nikdy splněna. Jako řešení rozšiřujícího zadání mohou žáci přidat do podmínky větve při jejím nesplnění a do ní nastavit na displeji nulu a zobrazit na pásku nějakou třetí barvu, při které budeme vidět, že metr neměřil hodnoty menší nebo rovné 90 cm.

Vzorové řešení rozšiřujícího zadání:

```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)
nastav 4digit na 4-Digit Display at P2 and P16

opakuj stále
nastav hodnota na Ultrasonic Sensor (in cm) at P1
když hodnota ≤ 90 tak
  4digit show number zaokrouhli hodnota
  nastav část1 na strip range from 0 with mapuj zaokrouhli hodnota z intervalu od 0 do 90 na interval od 0 do 29 leds
  nastav část2 na strip range from mapuj zaokrouhli hodnota z intervalu od 0 do 90 na interval od 0 do 29 with 30 leds
  část1 show color red
  část2 show color blue
jinak
  4digit show number 0
  strip show color green
```

Řešení rozšiřujících úloh

Úloha 1 – Rozsvěčíme pásek

```
při startu
nastav strip na NeoPixel at pin P0 with 30 leds as RGB (GRB format)
strip show color red 173 green 216 blue 230
```

Úloha 2 – Střídající se barvy

The image shows a Scratch script for controlling an LED strip. It starts with a 'when green flag clicked' event block. The first block is 'strip' (a red block) with 'show color' set to 'red'. This is followed by a 'wait' block (blue) for 500 ms. The sequence then repeats for 'orange', 'yellow', 'green', 'blue', and 'indigo', each with a 500 ms wait block. The script is contained within a 'when green flag clicked' event block.

```
when green flag clicked
  strip show color red
  wait 500 ms
  strip show color orange
  wait 500 ms
  strip show color yellow
  wait 500 ms
  strip show color green
  wait 500 ms
  strip show color blue
  wait 500 ms
  strip show color indigo
  wait 500 ms
```

Úloha 3 – Časovač

```
when green flag clicked
  configure strip on NeoPixel at pin P0 with 30 leds as RGB (GRB format)
  strip show color violet

when button A pressed
  repeat 4 times
    repeat 29 times
      strip shift pixels by 1
      strip show
      wait 1000 ms
    strip show color violet
    wait 1000 ms
```

The code is written in Scratch and is organized into two main event-driven sections. The first section, 'při startu' (when green flag clicked), sets up a NeoPixel strip on pin P0 with 30 LEDs in RGB (GRB) format and shows the color violet. The second section, 'při stisknutí tlačítka A' (when button A is pressed), contains a nested loop structure. It repeats 4 times, and within each iteration, it repeats 29 times. Inside the inner loop, the strip is shifted by 1 pixel, shown, and waited for 1000 ms. After the inner loop, the strip is shown as violet and waited for another 1000 ms.

Úloha 5 – Každá třetí zčervená

```
when green flag clicked
  configure strip on NeoPixel at pin P0 with 30 leds as RGB (GRB format)
  strip show color green
  set position to -1

repeat forever
  change position by 5
  strip set pixel color at position to red
  strip show
```

The code is written in Scratch and is organized into two main event-driven sections. The first section, 'při startu' (when green flag clicked), sets up a NeoPixel strip on pin P0 with 30 LEDs in RGB (GRB) format, shows the color green, and sets the position to -1. The second section, 'opakuj stále' (repeat forever), contains a loop that changes the position by 5, sets the pixel color at that position to red, and shows the strip.