

# Set Programátor Začátečník

Vítejte v návodu k naší sadě Programátor začátečník! Představujeme vám ideální volbu pro ty, kdo chtějí proniknout do světa programování a elektroniky. Tato sada obsahuje základní senzory, jako je IR senzor, fotorezistor a senzor barvy, které umožňují rozlišovat různé podněty z okolí. Mezi akční členy patří DC motor, OLED displej a programovatelný LED pásek "Neopixel", spolu s klasickými jednobarevnými LED. Sada dále obsahuje tlačítko a potenciometr, čímž poskytuje široké možnosti interakce.

Všechny součástky lze jednoduše připojit jedním kabelem k naší rozšiřující desce, která se řídí pomocí micro:bitu. Díky tomu je zapojení snadné a intuitivní. Sada je navržena tak, aby vás naučila základy programování v prostředí MakeCode a umožnila vám ovládat jednotlivé komponenty prostřednictvím jednoduchých a srozumitelných kroků.

Tato sada je perfektním nástrojem pro získání prvních zkušeností s programováním a elektronickými projekty, a poskytuje pevné základy pro další rozvoj vašich dovedností v oblasti technologie.

## Obsah balení

### Rozšíření

- Rozšiřující deska MB2
- Rozšíření pro IDC konektory

### Moduly

- více druhů

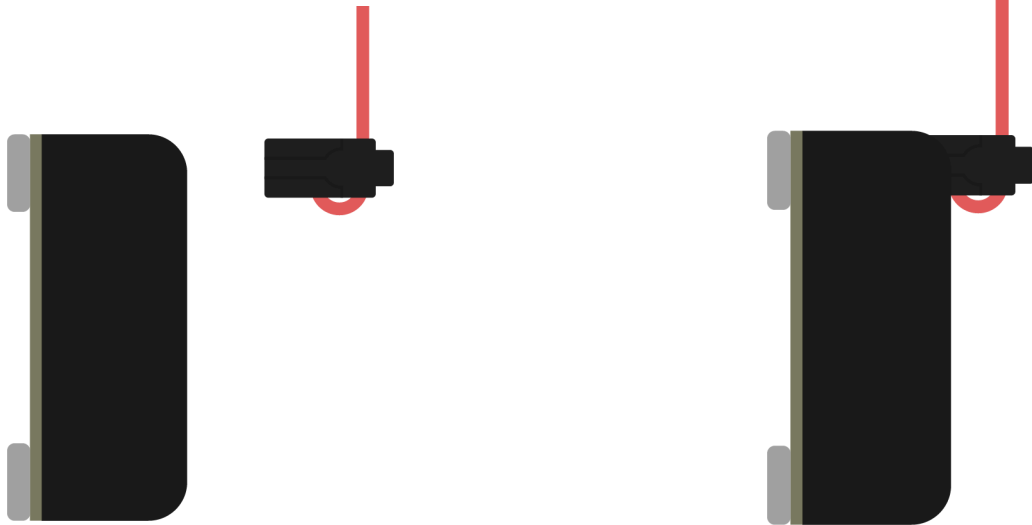
### Propojení

- 4x krátký kabel s IDC konektory
- 2x dlouhý kabel s IDC konektory

# Zapojení IDC

Díky použití IDC konektorů je zapojení bločků velmi jednoduché a pro všechny stejné. Nezáleží na typu komunikace daného bločku.

Stejné zapojení je použito na straně micro:bitu.



# Piny

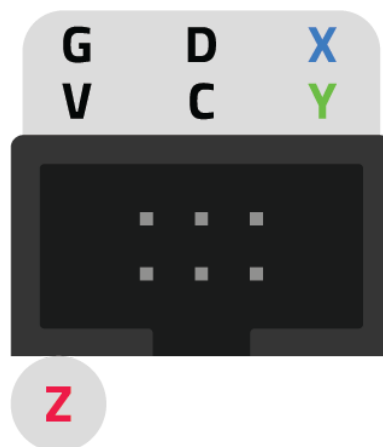
Pro vytvoření programu, který pracuje s bločky, je třeba vědět, které piny micro:bitu daný bloček využívá. V této kapitole si číslování vysvětlíme.

## Číslo konektoru **Z**

Toto číslo označuje pozici konektoru zleva. Budeme jej používat v programu v naší knihovně.

## Číslo pinu **X, Y**

Číslo X odpovídá číslu pinu micro:bitu. Toto číslo použijeme v programu pouze pokud chceme využít jiné knihovny.



## Piny napájení

**G** - Ground, zem

**V** - Voltage, napájení

## Piny komunikační (I2C)

**D** - Data

**C** - Clock, hodinový signál pro synchronizaci komunikace

# Programování

Programování micro:bitu provádíme v prostředí **MakeCode**. Toto prostředí obsahuje hotové funkce pro ovládání micro:bitu. Kromě vestavěných funkcí lze do prostředí MakeCode přidat nové funkce pomocí přidání tzv. knihovny.

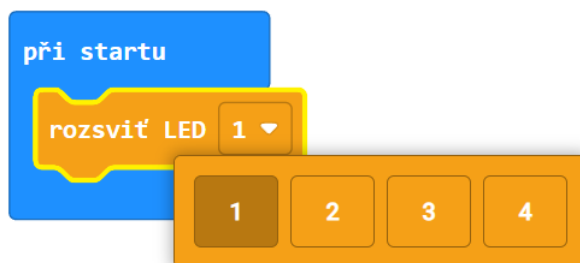
## Knihovna

V tomto návodu budeme používat naši vlastní knihovnu Začátečník. Knihovna je navržena přímo pro tuto sadu a naleznete v ní funkce pro ovládání všech bločků.

Tuto knihovnu můžete najít na našem webu v sekci Projekty - Práce se začátečníkem.

## Výběr konektoru

U většiny funkčních bloků je třeba vybrat, k jakému konektoru je daný modul připojen.

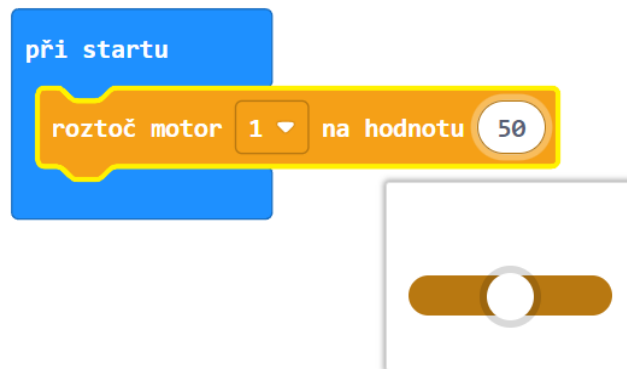


## Typy bloků

Bloky mohou mít různé způsoby využití. Jejich typy se liší i v tom, jak vypadají, tedy jejich tvarem. V naší knihovně používáme 3 základní typy: Funkce, Hodnota (číslo nebo text), Logická hodnota (pravda/nepravda).

## Funkce

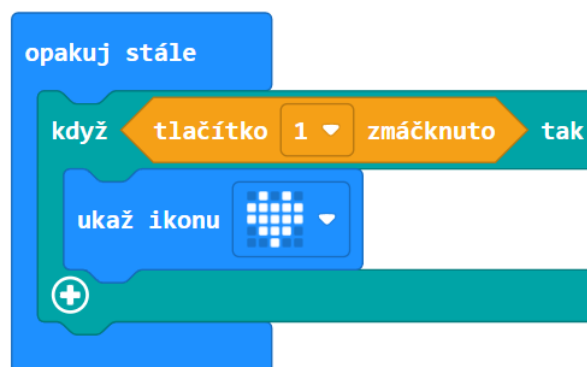
Prvním typem bloků v prostředí MakeCode je funkce. Vypadá stejně jako blok na předchozím obrázku. Funkce často požaduje zadání parametrů, například číslo konektoru, text, atd.



Obrázek ukazuje použití bloku pro roztočení motorku. Motorek je připojen na konektor 1 a chceme jej roztočit na číselnou hodnotu 50. Všechny číselné hodnoty v této knihovně nabývají hodnot 0-100.

## Logická hodnota

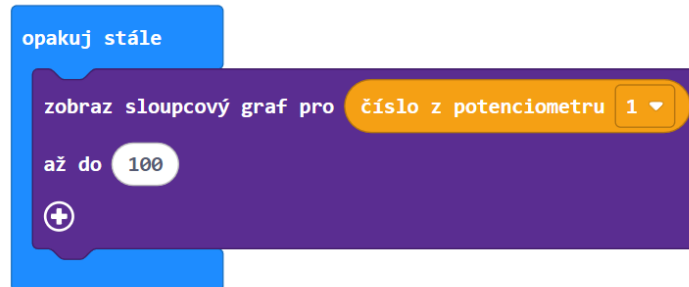
Druhým typem je logická hodnota. Blok je na stranách špičatý. Narozdíl od funkce se nedá použít v programu samotný, ale používá se právě jako parametr jiných funkcí.



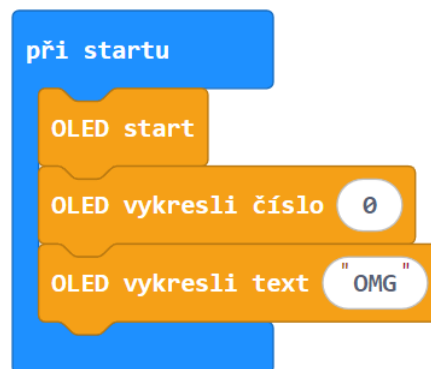
V tomto příkladě logickou hodnotu použijeme v podmínce. Když je blok "tlačítko 1 zmáčkuto" **pravdivý** (doopravdy je v tomto konektoru zapojeno tlačítko a je zmáčkuto), provede se část kódu uvnitř podmínky - na micro:bitu se zobrazí ikona srdce.

## Číselná hodnota nebo text

Poslední typ má kulatý tvar a může obsahovat různé hodnoty. Buď se může jednat o číslo, nebo text.



Zde čteme číselnou hodnotu z bločku Potenciometru připojeném ke konektoru 1 a toto číslo zapisujeme do funkce, která nám číslo zobrazí ve formě grafu na displeji micro:bitu.



Číselnou hodnotu nebo text můžeme zapsat i ručně, například při vypisování informací na OLED displej. Lze vidět, že parametry ve funkcích na obrázku mají stejně kulatý tvar, jako blok potenciometru na předchozím obrázku.

## Kategorie bloků

Většina modulků má své vlastní bloky pro jejich ovládání v programu. Ty jsou rozděleny do kategorií jako “Tlačítko”, “LED”, apod. Knihovna navíc obsahuje i pokročilejší bločky, které se skrývají pod tlačítkem “Více”.

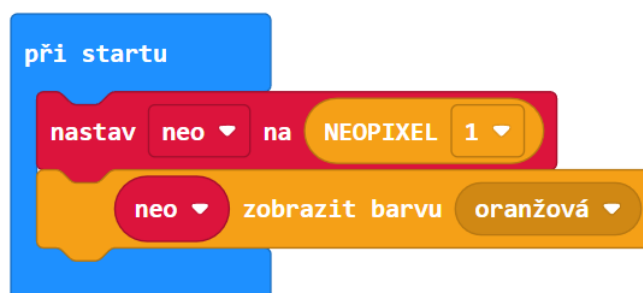
## Speciální bloky

Modulky **OLED**, **BME**, **Senzor barvy** využívají pro přenos informací speciální typ komunikace. Není u nich třeba vybírat číslo konektoru, ale je nutné na začátku programu jejich komunikaci zapnout - blok **start**.



## Neopixel bloky

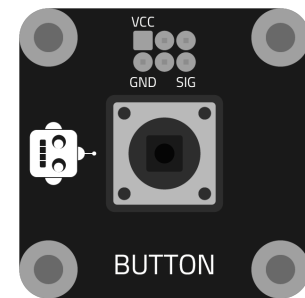
Tento modulek vyžaduje odlišné ovládání od všech ostatních modulků. Místo bloku **start** je třeba vytvořit proměnnou **neo**, která obsahuje všechny potřebné informace o použitém modulku.



Následně již opět není třeba zadávat číslo konektoru, ale stačí se odkazovat na danou proměnnou **neo**, jako například při rozsvícení červené barvy.

# Tlačítko

Tlačítko je nejjednodušší ovládací prvek. Stisknutí tlačítka představuje spojení (uzavření) elektrického obvodu, stejně jako když spojíme 2 drátky k sobě. To umožní průchod elektrického proudu do vstupu micro:bitu. Tlačítkem můžeme ovládat všechny další bločky v balení, jako například Neopixel, motorek, nebo jednotlivá barevná světla.



Tlačítko představuje vstup, který má pouze 2 úrovně - ZAPNUTO, VYPNUTO. Takovému vstupu říkáme logický.

## Knihovna

Pro tlačítko je připraven blok "tlačítko Z zmáčknuť". Stačí vybrat číslo konektoru Z. Výstupem je logická hodnota, tzn. pravda/nepravda. Pokud tedy tlačítko zmáčknuť, blok získá hodnotu pravda.

### Více

Pokročilá funkce má místo výstupu *pravda/nepravda* číselný výstup 1/0.

## Programování

Ukážeme si, že tlačítko slouží jako základní ovládací prvek. Pro následující úlohy zapojíme tlačítko do **konektoru 1**.

### Rozsvícení displeje

[https://makecode.microbit.org/\\_JAyCeMPC5U3E](https://makecode.microbit.org/_JAyCeMPC5U3E)

V bločku opakuj stále kontrolujeme podmínku, zda je tlačítko stisknuto. Pokud je, zobrazíme na displeji ikonu. Pokud ne, displej vymažeme.

### Hrací kostka

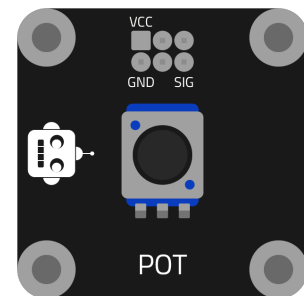
[https://makecode.microbit.org/\\_1JdfhtR4bCt8](https://makecode.microbit.org/_1JdfhtR4bCt8)

V tomto příkladu nepotřebujeme stav uvolněného tlačítka využívat. Stačí nám v případě stisknutého tlačítka zobrazit číslo na displeji. K vygenerování čísla využijeme matematickou knihovnu, která obsahuje funkci "náhodné číslo". Interval čísel volíme stejný, jako je na klasické hrací kostce.



# Potenciometr

Potenciometr je speciální druh rezistoru, který se používá jako snímač. Pomocí něj můžeme řídit elektrické obvody, především ovládat hlasitost, jas, rychlost a další parametry. Potenciometr je v principu rezistor, jehož odpor můžeme upravit otočením hlavičky. Změnou úrovně odporu se mění proud protékající obvodem, což opět snímá micro:bit na svém vstupním pinu.



Potenciometr nabývá více úrovní, a jde o **analogový signál**. Vstupní signály micro:bitu však mají předem dané rozlišení. Základní knihovna “piny” rozlišuje hodnoty 0-1023. Naše knihovna pracuje s hodnotami 0-100.

**Pozor!** Potenciometr nefunguje v konektoru 4.

## Knihovna

Pro potenciometr je připraven 1 blok. Ten nabývá číselné hodnoty podle toho, jak potenciometr natočíme. Hodnoty se nachází v rozmezí 0-100.

## Programování

Potenciometr představuje pokročilejší ovládací prvek, jelikož nabývá více hodnot. Pro tyto příklady zapojíme potenciometr do **konektoru 1**.

### Zobrazení hodnoty potenciometru

[https://makecode.microbit.org/\\_Cmk0aiXz01xa](https://makecode.microbit.org/_Cmk0aiXz01xa)

Pro grafické zobrazení měřené hodnoty lze využít funkci “zobraz sloupcový graf”. Zobrazovanou hodnotou je právě blok . Jako maximální hodnota je číslo 1023 (to je dáno převodníkem, který je zabudovaný v micro:bitu).

## **Nastavení frekvence blikání**

[https://makecode.microbit.org/\\_AFHFsjAM9Lr1](https://makecode.microbit.org/_AFHFsjAM9Lr1)

Místo toho, abychom hodnotu zobrazovali na displeji, můžeme ji použít na nastavení frekvence (rychlosti) blikání. Využijeme blok “přepni x y” z knihovny LED. Tento blok rozsvítí konkrétní diodu v případě, že je zhasnutá. Pokud již svítí, zhasne ji.

Následně přidáme do programu blok “čekej x ms”. Místo jednoho čísla zapíšeme do tohoto bloku hodnotu na potenciometru.

# LED světlo

LED je součástka, která vyzařuje světlo dané barvy. Dnes se používá téměř všude, jelikož je velmi levná, energeticky úsporná a má dlouhou životnost. LED můžeme ovládat jak logickým výstupem a zapsat do něj hodnoty **1 a 0**, tak i “analogovým” a zapisovat hodnotu intenzity svícení.



## Knihovna

Pomocí knihovny můžeme LED rozsvítit funkcí “rozsviť LED”, nebo zhasnout pomocí “zhasni LED”. Navíc může LED svítit danou intenzitou pomocí bloku “zapiš do LED číslo”.

### Více

LED má navíc připraveny i pokročilé bloky. “Přepni LED” zapne danou LED v případě, že nesvítí a naopak vypne, pokud svítí. Blok “Rozsviť LED? pravda/nepravda” zapíná LED v případě, že do něj vložíme hodnotu *pravda*.

## Programování

LED zapojíme do **konektoru 2**.

### Kontrolka

[https://makecode.microbit.org/\\_RetWMKarTDMp](https://makecode.microbit.org/_RetWMKarTDMp)

Do **konektoru 1** zapojíme tlačítko. Stejně jako v kapitole pro tlačítko, použijeme strukturu s podmínkou “když x tak A, jinak B”. Do podmínky použijeme hodnotu tlačítka. Pokud je zmáčknuto, rozsvítíme LED pomocí bloku “rozsviť LED”. Nesmíme zapomenout na změnu čísla konektoru. Pro zhasnutí LED použijeme blok “zhasni LED”.

### Nastavení intenzity

[https://makecode.microbit.org/\\_TfHLCh562Via](https://makecode.microbit.org/_TfHLCh562Via)

Do **konektoru 1** zapojíme místo tlačítka potenciometr. Struktura programu je velmi jednoduchá. Do LED chceme zapsat číslo, které představuje intenzitu svícení. Toto číslo získáme z bloku “číslo z potenciometru”.

## Nastavení frekvence blikání

[https://makecode.microbit.org/\\_EAKAJWLM716e](https://makecode.microbit.org/_EAKAJWLM716e)

Pro pravidelné blikání LED využijeme blok “přepni LED”. Ten se nachází v kategorii pokročilejších bloků (pod knihovnou se nachází tlačítko “více”). Pokud LED svítí, zavoláním této funkce se LED zhasne. Naopak pokud je LED zhasnutá a zavoláme tuto funkci, LED se rozsvítí.

Dalším krokem je nastavení pauzy programu, tedy použití bloku “čekej x ms”. Zde vložíme blok “číslo z potenciometru”. Nezapomeňme modulům přiřadit správná čísla konektoru.

## Nastavení frekvence blikání (pokročilé)

[https://makecode.microbit.org/\\_eD1UtzMy0M23](https://makecode.microbit.org/_eD1UtzMy0M23)

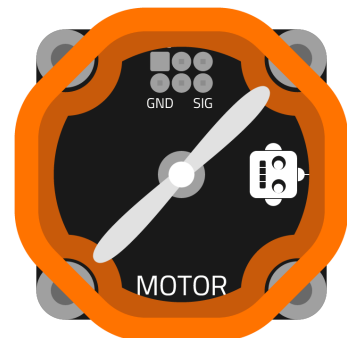
Pokud nechceme využít blok “přepni LED”, můžeme si tuto logiku naprogramovat sami pomocí podmínky. Abychom mohli zjistit, zda LED svítí nebo ne, potřebujeme si tento stav ukládat do proměnné. Vytvoříme si tak proměnnou o hodnotě nepravda. To znamená, že na začátku programu nic nesvítí. Následně budeme ve smyčce zjišťovat, zda chceme LED rozsvítit, nebo zhasnout. Pokud LED svítí, tzn. proměnná “svítí” je *pravda* (svítí = pravda), budeme chtít LED zhasnout pomocí bloku “zhasni LED”. Zároveň tuto změnu stavu zapíšeme do naší proměnné “svítí”. Nastavíme tedy “svítí” na *nepravda*. V opačném případě LED rozsvítíme a do proměnné zapíšeme *pravda*.

Nakonec opět nastavíme pauzu programu pomocí “čekej x ms”, kde použijeme hodnotu potenciometru.

Opět nezapomínejme všude nastavit správná čísla konektoru.

# Motorek

Vrtulka je poháněna stejnosměrným motorem (DC motorem). Takový motor obsahuje magnet - **stator**, který působí na pohyblivou část motoru - **rotor**. Jakmile cívkou na rotoru začne procházet elektrický proud, vzniká magnetické pole. To reaguje s magnetickým polem magnetů statoru a vzniká síla otáčející rotorem.



DC motor se ovládá analogovým signálem. Podle úrovně napětí na svorkách motoru se motor otáčí danou rychlostí. K nastavení otáček tak můžeme použít i potenciometr.

## Knihovna

Bloky pro motorek jsou v principu stejné jako bloky pro LED. Motorek můžeme zapnout, nebo vypnout. Navíc lze nastavit jeho otáčky na číselnou hodnotu 0 až 100.

### Více

Stejně jako LED motorek můžeme přepínat jedním blokem, nebo zapínat podle zadané logické hodnoty *pravda/nepavda*.

## Programování

Motorek zapojíme do **konektoru 2**.

### Spínání motorku tlačítkem

[https://makecode.microbit.org/\\_hKrXVMApCeh7](https://makecode.microbit.org/_hKrXVMApCeh7)

Do **konektoru 1** zapojíme tlačítko. Pokud je tlačítko zmáčknuo, roztočíme motor. Jinak motor vypneme.

### Nastavení otáček potenciometrem

[https://makecode.microbit.org/\\_PX4cr2iujF5m](https://makecode.microbit.org/_PX4cr2iujF5m)

Do **konektoru 1** zapojíme potenciometr. Opět použijeme strukturu jako u LED. Hodnotu z potenciometru zapíšeme do bloku pro roztočení motoru.

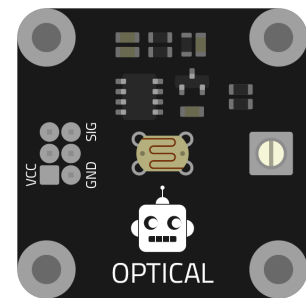
# Světelný senzor

Tento senzor je schopen snímat intenzitu dopadajícího světla. Díky tomu jsme schopni měřit spoustu věcí. Jednak zda je den či noc, zda máme v místnosti rozsvícené světlo, zda se před ním nachází překážka a tak dále.

Hlavním členem senzoru je **fotorezistor**. Podobně jako potenciometr se jedná o speciální druh rezistoru, který mění svou hodnotu odporu.

V tomto případě je odpor závislý na intenzitě dopadajícího světla. Tím se opět změní hodnota proudu procházejícího obvodem a tu opět měří micro:bit na svém vstupním pinu.

Na přední straně bločku se nachází také potenciometr. Pomocí něj provádíme kalibraci. Pro měření ve tmě totiž potřebujeme jinou citlivost, než pro měření za světla.



## Knihovna

V případě senzorů “Světelný senzor, IR senzor a UV senzor” používáme stejné bloky. Jejich výstupem je pravdivostní hodnota *pravda/nepravda*. V pokročilých blocích existuje stejný blok, akorát s výstupem 1/0.

## Programování

Senzor zapojíme do **konektoru 1**.

### Automatický ventilátor

[https://makecode.microbit.org/\\_5dC0L89Ty91U](https://makecode.microbit.org/_5dC0L89Ty91U)

Motor zapojíme do **konektoru 2**. Struktura je opět stejná, jako u spínání motoru tlačítkem. Místo hodnoty z tlačítka používáme hodnotu ze senzoru.

### Výrobní linka

[https://makecode.microbit.org/\\_YH35tM2AgRik](https://makecode.microbit.org/_YH35tM2AgRik)

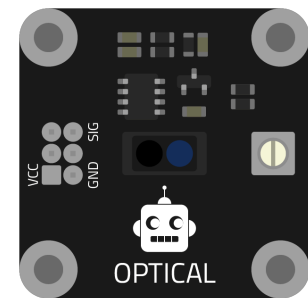
Pokud senzor uvidí na páse překážku, rozsvítí se červená LED. Jinak svítí zelená.

# IR senzor

Tento senzor je schopen snímat infračervené (IR) záření, které je pro lidské oko neviditelné. Díky tomu jsme schopni detekovat teplo, pohyb nebo přítomnost objektů, které vyzařují nebo odrážejí IR světlo.

Hlavním členem senzoru je infračervená dioda a detektor. Dioda vysílá IR paprsky, které se odrazí od objektu a vrací se zpět k detektoru. Když detektor zaznamená odražené světlo, změní se hodnota elektrického signálu, kterou micro:bit vyhodnotí na svém vstupním pinu.

Na senzoru je opět potenciometr, kterým nastavujeme měřenou úroveň. Pokud chceme zjišťovat objekt dále od senzoru, potřebujeme citlivější nastavení.



## Knihovna

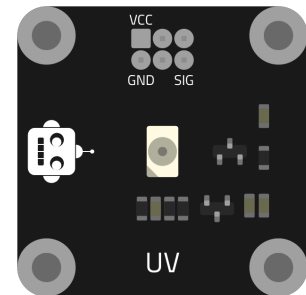
V případě senzorů “Světelný senzor, IR senzor a UV senzor” používáme stejné bloky. Jejich výstupem je pravdivostní hodnota *pravda/nepravda*. V pokročilých blocích existuje stejný blok, akorát s výstupem 1/0.

## Programování

Příklady úkolů jsou stejné, jako u předešlého senzoru.

# UV senzor

Tento senzor je schopen snímat ultrafialové (UV) záření, které je pro lidské oko opět neviditelné. Takové záření přichází například ze slunce. Díky tomu jsme schopni měřit, jak silné UV záření dopadá na senzor, což může být užitečné pro monitorování bezpečnosti na slunci nebo detekci UV světelných zdrojů.



Hlavním členem senzoru je speciální fotodetektor citlivý na UV záření. Když na něj dopadá UV světlo, generuje elektrický signál. Tím se změní hodnota proudu procházejícího obvodem a tuto změnu opět měří micro:bit na svém vstupním pinu.

Na senzoru je opět potenciometr, kterým nastavujeme měřenou úroveň. Pokud chceme zjišťovat objekt dále od senzoru, potřebujeme citlivější nastavení.

## Knihovna

V případě senzorů “Světelný senzor, IR senzor a UV senzor” používáme stejné bloky. Jejich výstupem je pravdivostní hodnota *pravda/nepravda*. V pokročilých blocích existuje stejný blok, akorát s výstupem 1/0.

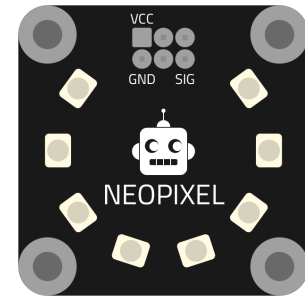
## Programování

Příklady úkolů jsou stejné, jako u Světelného senzoru.



# Neopixel

Tento bloček obsahuje 8 světýlek (pro nás tzv. pixelů). Jedná se o speciální LEDky, které dokážou svítit jakoukoliv barvou. Označují se zkratkou RGB, jelikož jedna taková součástka uvnitř skrývá 3 klasické LED, které svítí červenou (R - red), zelenou (G - green) a modrou (B - blue) barvou. Kombinací těchto barev získáme jakoukoliv barvu a jakoukoliv úroveň jasů. Na tomto principu pracují některé obrazovky mobilních telefonů, počítačů a televizorů.



RGB LED pásy jsou v dnešní době populární a mají mnoho způsobů využití. My jej můžeme použít například pro zobrazení rychlosti otáčení motoru, natočení potenciometru nebo hodnoty jiných senzorů.

## Knihovna

Pro ovládání Neopixelu je nutné na začátku programu vytvořit proměnnou, která obsahuje všechny potřebné informace, jako například číslo konektoru. Poté nám stačí se odkazovat pouze na tuto proměnnou. V základních funkcích jsou bloky k zobrazení barvy, duhy a sloupcového grafu.

## Programování

Neopixel připojíme do **konektoru 2**.

### Nastavení pro použití Neopixel

[https://makecode.microbit.org/\\_K2zd70LgCbM8](https://makecode.microbit.org/_K2zd70LgCbM8)

Nejprve si ukážeme, jak je třeba Neopixel nastavit na začátku kódu. V kategorii Neopixel vybereme první blok. Vidíme, že do proměnné "neo" uložíme objekt NEOPIXEL a zvolíme konektor 2. Tím máme vytvořený objekt "neo", pomocí kterého budeme volat další funkce. Vyzkoušíme si základní zobrazení barvy pomocí "neo zobrazit barvu" a barvu si zvolíme.

## Zobrazení hodnoty potenciometru

[https://makecode.microbit.org/\\_RsKDmhVcJ8dk](https://makecode.microbit.org/_RsKDmhVcJ8dk)

Místo zobrazení grafu na displeji micro:bitu můžeme použít právě Neopixel. Díky funkci “zobraz sloupcový graf” můžeme hodnotu potenciometru ukázat přehlednějším způsobem. Opět nejprve odkážeme na objekt “neo”, poté vložíme zobrazovanou hodnotu, tedy “číslo z potenciometru” a do poslední buňky “až po” vepíšeme hodnotu 100 (původní hodnota), jelikož právě to je maximum, které z potenciometru čteme.

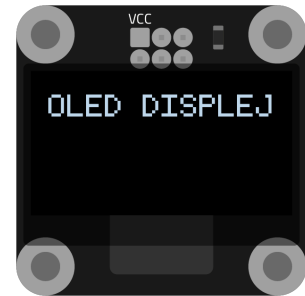
## Plynulý přechod barvy (pokročilý)

[https://makecode.microbit.org/\\_a3gHX88pjd0K](https://makecode.microbit.org/_a3gHX88pjd0K)

Tento program plynule mění odstín světla. Použijeme k tomu pokročilé funkce Neopixelu, tedy výběr barvy ve formátu HSL, tedy zjednodušeně odstín, sytost a jas. Nejprve si vytvoříme proměnnou odstín a nastavíme ji na 0. Ve smyčce použijeme blok “neo zobrazit barvu” a na místo přednastavené barvy vložíme blok “odstín H sytost S jas L”. Do první buňky, která reprezentuje odstín, vložíme naši proměnnou. Sytost nastavíme na 99 (tedy maximální hodnotu) a jas na 49 (tedy střed rozsahu). Každý cyklus programu přičteme k proměnné odstín hodnotu 1. Na začátek pak musíme přidat podmínku, která nám zajistí, že při dosažení maximální hodnoty odstínu (360) se proměnná opět vynuluje.

# OLED displej

Tento displej je schopen zobrazovat text nebo jednoduchou grafiku pomocí jednotlivých světelných bodů, které se rozsvěcují přímo na obrazovce. Díky tomu můžeme zobrazit různé informace, jako jsou čísla, texty nebo jednoduché obrázky.



Každý pixel displeje je tvořen malou LED diodou, která se rozsvítí podle signálu z micro:bitu, čímž se vytváří požadovaný obraz na displeji.

## Knihovna

Pro ovládání **OLED displeje, BME a Senzoru barvy** opět použijeme jiný přístup, než pro předchozí modulky. Modulky komunikují pomocí protokolu I2C a používají k tomu jiné piny. Není tak třeba nastavovat, ke kterému konektoru je připojíme, ale je třeba tuto komunikaci zapnout blokem **start**.

## Programování

OLED zapojíme do **konektoru 4**. Na začátku každého programu použijeme blok **OLED start**. Pokud chceme ve smyčce vypisovat text nebo čísla, musíme vždy zaplnit všech 8 řádků. K tomu využíváme blok "OLED vykresli x prázdných řádků".

### Pozdrav

[https://makecode.microbit.org/\\_HJ0FPCTWoAE3](https://makecode.microbit.org/_HJ0FPCTWoAE3)

Pro otestování funkce OLED si vypíšeme na displej pozdrav.

### Zobrazení číselné hodnoty

[https://makecode.microbit.org/\\_eKwHR6VmgE2u](https://makecode.microbit.org/_eKwHR6VmgE2u)

Potenciometr připojíme ke **konektoru 1**. V tomto programu vypisujeme nejprve text "potenciometr", poté vynecháme 1 řádek, pak píšeme číselnou hodnotu z potenciometru a nakonec vynecháme 5 řádků.

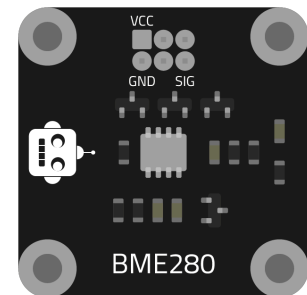
## Zápis textu a čísla do 1 řádku

[https://makecode.microbit.org/\\_8X9EHAWzCTwW](https://makecode.microbit.org/_8X9EHAWzCTwW)

Potenciometr připojíme ke **konektoru 1**. Abychom zapsali text i číselnou hodnotu do jednoho řádku, využijeme pouze blok “OLED vykresli text”, do kterého vložíme z knihovny **Text** blok “spoj x y”. Do kolonky x vložíme text “hodnota: “ s mezerou na konci, do kolonky y vložíme číslo, které chceme zobrazit. Tato funkce sama převede číselnou hodnotu na text. Zbytek kódu je stejný, jako předchozí příklad.

# BME meteostanice

Tento senzor je schopen měřit základní meteorologické veličiny, jako jsou teplota, vlhkost vzduchu a atmosférický tlak. Díky tomu můžeme sledovat změny počasí a vytvářet různé projekty, například domácí meteorologickou stanici.



Hlavním členem senzoru je speciální čip, který obsahuje senzory pro měření teploty, vlhkosti a tlaku. Každý z těchto senzorů přeměňuje naměřené hodnoty na elektrický signál, který následně micro:bit zpracuje na svém vstupním pinu a převede na čitelné hodnoty, které můžeme zobrazit například na OLED displeji.

## Knihovna

Pro ovládání **OLED displeje, BME a Senzoru barvy** opět použijeme jiný přístup, než pro předchozí modulky. Modulky komunikují pomocí protokolu I2C a používají k tomu jiné piny. Není tak třeba nastavovat, ke kterému konektoru je připojíme, ale je třeba tuto komunikaci zapnout blokem **start**.

## Programování

BME zapojíme do **konektoru 3**. Na začátku každého programu použijeme blok BME start.

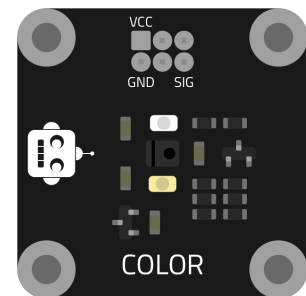
### Vypsání hodnot na OLED displej

[https://makecode.microbit.org/\\_Mq1Xzdguk0wx](https://makecode.microbit.org/_Mq1Xzdguk0wx)

OLED displej připojíme ke **konektoru 4**. Jednoduše vypíšeme do všech 8 řádků požadované hodnoty. Nejprve předepíšeme název měřené veličiny spolu s jednotkou a následně číselnou hodnotu.

# Senzor barvy

Tento senzor je schopen rozpoznávat barvy a měřit intenzitu světla, díky čemuž můžeme detekovat různé barvy objektů a vytvářet interaktivní projekty, jako je rozpoznávání barevného kódu nebo třídění objektů podle barvy.



Hlavním členem senzoru je speciální foto detektor, který měří intenzitu světla v různých částech spektra (červená, zelená, modrá a infračervená). Senzor APDS-9960 vyhodnocuje tyto údaje a micro:bit je následně zpracovává na svém vstupním pinu, což nám umožňuje určit, jakou barvu detekuje.

## Knihovna

Pro ovládání **OLED displeje, BME a Senzoru barvy** opět použijeme jiný přístup, než pro předchozí modulky. Modulky komunikují pomocí protokolu I2C a používají k tomu jiné piny. Není tak třeba nastavovat, ke kterému konektoru je připojíme, ale je třeba tuto komunikaci zapnout blokem **start**.

## Více

Pokud chceme rozsvítit přídatnou LED na tomto modulku, potřebujeme zadat číslo konektoru.

## Programování

Senzor barvy zapojíme do **konektoru 3**. Na začátku každého programu použijeme blok COLOR start.

## Vypsání barevných složek na OLED displej

[https://makecode.microbit.org/\\_4w674y9UrdU4](https://makecode.microbit.org/_4w674y9UrdU4)

OLED displej připojíme ke **konektoru 4**. Kromě bloku "COLOR start" využijeme také "COLOR rozsvít přídatnou LED 3", jelikož senzor je připojen ke konektoru 3. Přídatná LED nám umožní měřit barvu objektů v blízkosti senzoru. Pokud tak přiblížíme k senzoru předmět, který má červenou barvu, bude hodnota R (red) nejvyšší. Zápis na displej je vysvětlen v kapitole OLED.

## **Teplota odraženého světla**

[https://makecode.microbit.org/\\_TXeAAe9tc75T](https://makecode.microbit.org/_TXeAAe9tc75T)

Ke konektoru 4 zapojíme Neopixel. Pokud je světlo dopadající na senzor teplé barvy, bude mít vyšší červenou složku. Pokud je světlo studené, bude modrá složka vyšší. V programu použijeme stejnou logiku a teplotu světla zobrazíme na Neopixelu. Pokud je hodnota červené barvy vyšší než hodnota modré, rozsvítí se červená na Neopixelu. V opačném případě svítí modrá.

# Specifikace

## Tlačítko

- 12 x 12 x 7,3 mm

## Potenciometr

- úhel: 300°
- hodnota: 10kΩ
- výkon: 50mW

## LED světlo

- průměr: 8mm

## Neopixel

- Napájení: 3V
- Stupeň krytí: IP30
- Výkon max: 18W/m
- Zobrazovací formát: RGB

## Motorek

- Napájení: 3 V
- Otáčky: 48 000 RPM
- Konstrukce: Permanentní magnet

## Světelný senzor

- čidlo: fotorezistor
- komparátor: LM393

## IR senzor

- čidlo: IR dioda, IR fotodioda
- komparátor: LM393
- detekční rozsah: 1 ÷ 15mm

## UV senzor

- výstup: analogový

## OLED displej

- rozlišení: 128 x 64px
- rozhraní: I2C

## BME

- rozsah teploty: -40 ÷ +85°C
- rozsah tlaku: 30 ÷ 110kPa
- rozsah vlhkosti: 0 ÷ 100%

## Senzor barvy

- senzor: RGB
- rozhraní: I2C