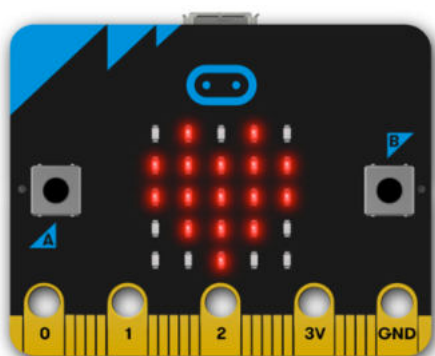


DigiMe Rozšiřující workshop

Interreg



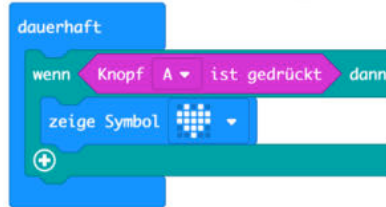
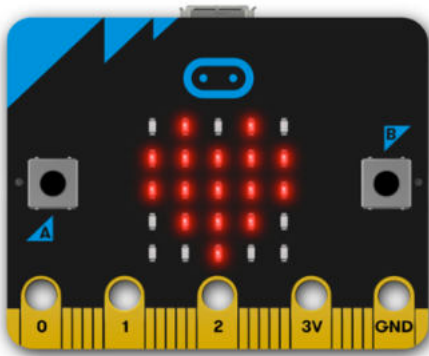
Österreich-Tschechische Republik

DigiMe

Europäischer Fonds für regionale Entwicklung

Project co-funded by the European Regional Development Fund (ERDF)

DigiMe Rozšiřující workshop



Průběh workshopu

► Obsah

- ◆ Stručný úvod (WH) micro:bit a makecode rozhraní
- ◆ Seznámení s rádiovým modulem (odesílání zpráv)
- ◆ Making-> Kódování a technologie (elektronika+senzory)

► Příklady a použití ve výuce

► Konstrukce mikrobitů (piny, napájení,...)

Praktická část

► Použití pinů mikrobitu:

- ◆ Výstup (LED diody, servo, motor, LC displej)
- ◆ Vstup (joystick, tlačítko)
- ◆ Senzory (světlo, pohyb, vzdálenost,...)

► Základy elektroniky (3 volt vs. 5 volt, řídící jednotka motoru,

► Použití a možnosti rádiového modulu

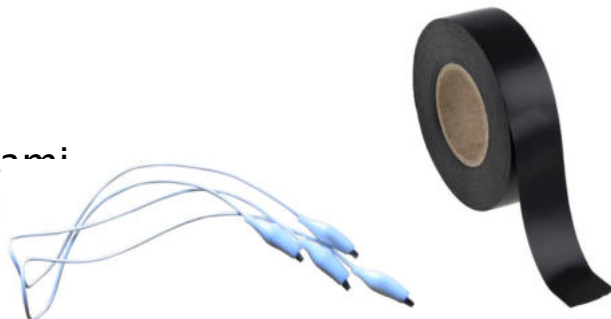
► Příklady možných projektů

Horký drát – konstrukce 1

Určitě znáš vědomostní hru „Horký drát“. Chceme tuto hru napodobnit pomocí mikrobitu, trochy elektroniky a modelářství. Podle toho, jak jsi pokročilý v kódování, můžeš přidat své vlastní funkce nebo rozšíření.

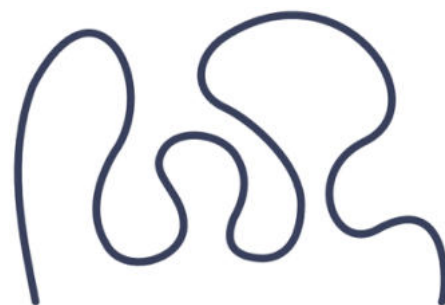
Co potřebuješ

- 1 kus drátu
- 2 kabely s krokodýlími svorkami
- 1 dřevěná deska nebo plast
- 1 micro:bit
- izolační páska



Konstrukce dráhy z drátu

Pomocí kleští ohni drát podle svého vkusu. Čím užší je dráha z drátu, tím obtížnější je zvládnout parkour bez chyby.

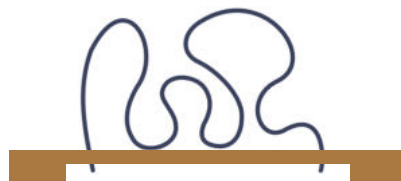


Na dřevěné desce nakresli počáteční a koncový bod dráhy z drátu a vyvrtej otvor o průměru drátu.

Tip: Pokud snížíš vzdálenost mezi otvory o 10 mm, dráha z drátu se lépe sevře v desce.

Nyní vložíš dráhu z drátu skrz otvory, které jsi právě vyvrtal. Poté ohni oba konce drátu (asi 10 mm) na spodní straně dřevěné desky a přebytečné konce odřízni boční frézou.

Namontuj 2 úzké dřevěné proužky na spodní stranu dřevěné desky tak, aby hra mohla stát naplocho na stole.



Alternativně můžeš použít 2 kusy plastelíny namísto dřevěného stojanu, kam natiskneš 2 konce dráhy z drátu.

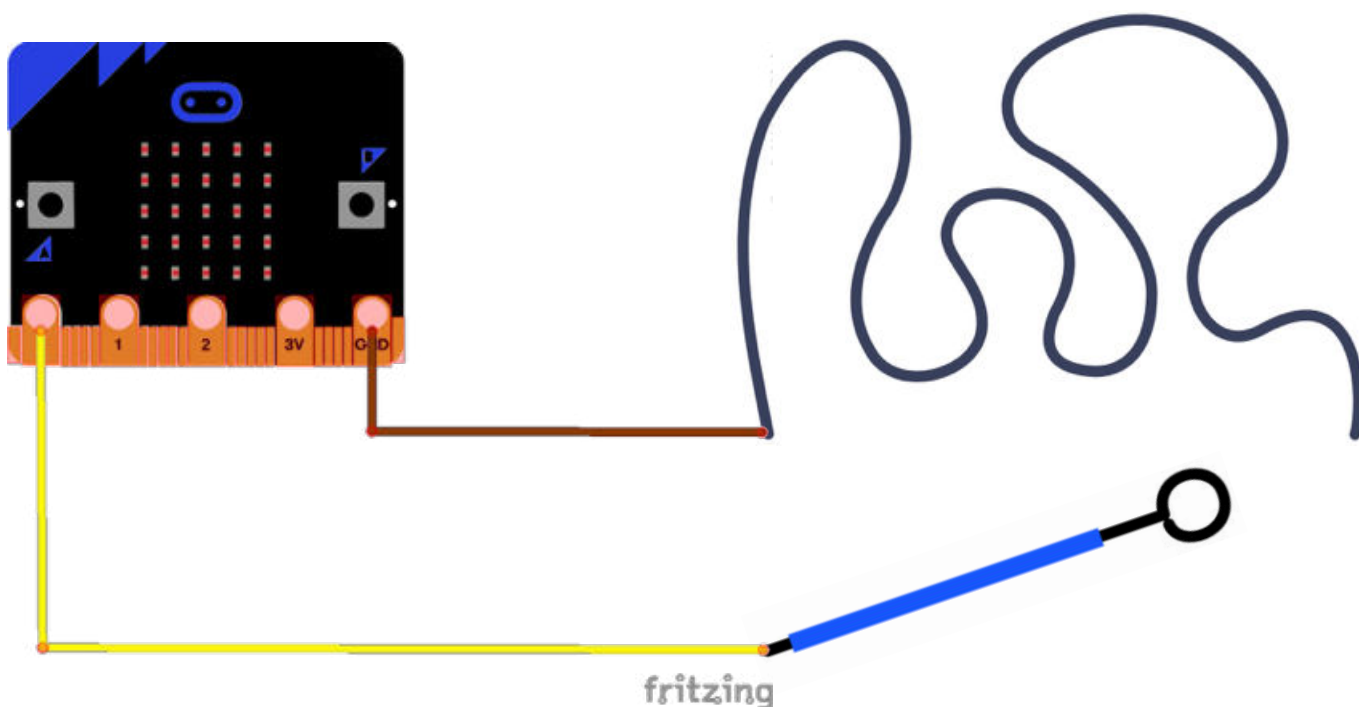
Horký drát – konstrukce 2

Konstrukce drátěné smyčky

Na konci kusu drátu o délce asi 15 cm ohni smyčku o průměru asi 15 mm. Čím menší je průměr, tím vyšší je obtížnost hry. Potom zabalíš hřídel izolační páskou, ale posledních 10 mm hřídele zůstává neizolovaných.

Propojení

Připojte jeden konec dráhy z drátu k záporné svorce (GND) mikrobitu pomocí kabelu krokodýlí svorky. Poté ved' druhý kabel krokodýlí svorky od pin0 mikrobitu k neizolovanému hřídeli drátěné smyčky.



Horký drát – kódování

Programování

Vzhledem k tomu, že tvoje drátová smyčka je připojena k Pin0, musíš se Pin0 dotázat, zda byl detekován dotyk.

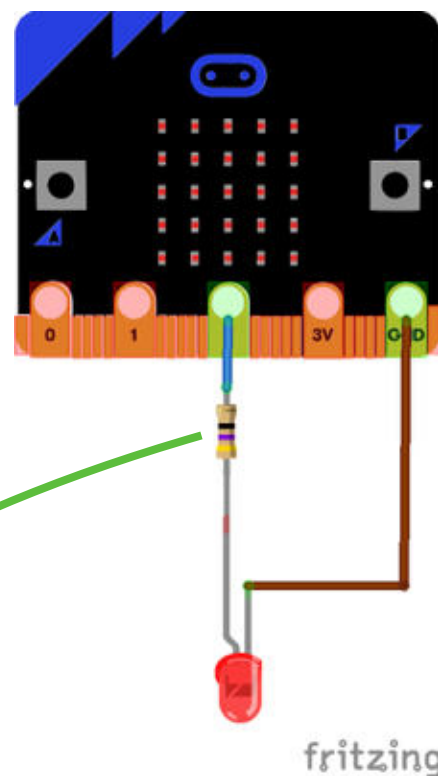
Pokud tomu tak je, zobrazí se smutný smajlík a poté se znovu smaže.



https://makecode.microbit.org/_a44akyhLFPyH

Chybové hlášení pomocí LED (volitelné)

Kromě toho si můžeš nechat zobrazit jakoukoli chybu pomocí zasvícení LEDky (světelná dioda). Vše, co potřebuješ, je světelná dioda v barvě podle tvého výběru a předřazený odpor (náklady jsou nižší než 1 EUR).



Upozornění

Aby se LED dioda během provozu nezničila, potřebuješ tzv. předřazený odpor. I když se takto nazývá, nezáleží na tom, zda je instalován v obvodu před nebo po světelné diodě. Pro „konvenční“ LED diody dostupné na trhu potřebuješ pro náš mikrobitový 3V zdroj napětí 47 ohmový rezistor s barvami černá–fialová–žlutá.

Další podrobnosti týkající se výpočtu předřazených odporů nalezneš v pracovním listu „Ohmův zákon“.

Horký drát – léto

Rozšíření léto



https://makecode.microbit.org/_LfhCLTHuXPY3

Tip

Blok

schreibe digitalen Wert von Pin P0 auf 0

pinu P1 na 1 najdeš v kategorii

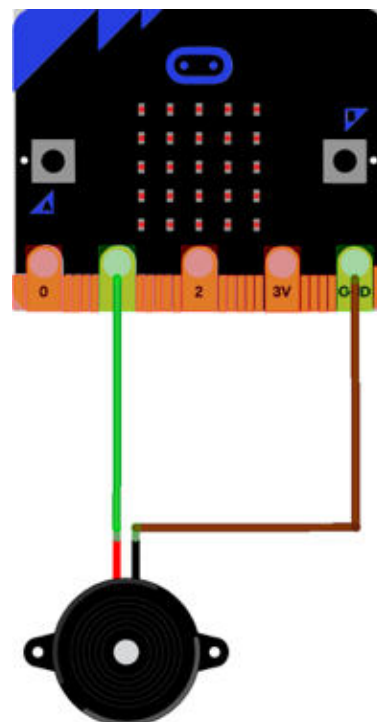
Pins

napiš digitální hodnotu
v části Pokročilí.

Fortgeschritten

Pro zvukovou indikaci můžete použít minibuzzer. Připoj záporný pól (krátký pin) pomocí kabelu krokodýlí svorky k GND mikrobitu, resp. k dráze z drátu (protože je již připojena k GND).

Pozitivní pól bzučáku (dlouhý pin) připojíš k pinu 1 mikrobitu pomocí jiného kabelu krokodýlí svorky.



fritzing

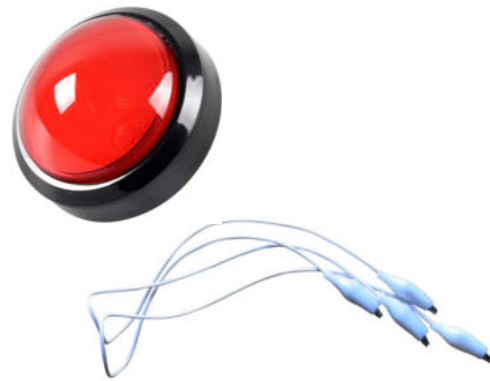
Hrát Bzučák – Konstrukce

Kdo tu situaci nezná? Několika hráčům je položena otázka a je nutné rozhodnout, kdo odpověděl jako první.

Toto rozhodnutí je nyní převzato samočinným naprogramovaným obvodem. U osoby, která zmáčkne tlačítko jako první, se objeví světlo a všichni ostatní hráči již nemají příležitost tlačítka stisknout. Teprve po uvolnění ze strany rozhodce bude vyhodnoceno nové stisknutí.

Co potřebuješ

- tlačítko
- 2 kabely s krokodýlími svorkami
- 1 micro:bit



Funk

Aby všichni hráči mohli komunikovat mezi sebou bezdrátově, je nutné použít rádiový modul, který je zabudován do mikrobitu.

beim Start

setze Funkgruppe auf 7

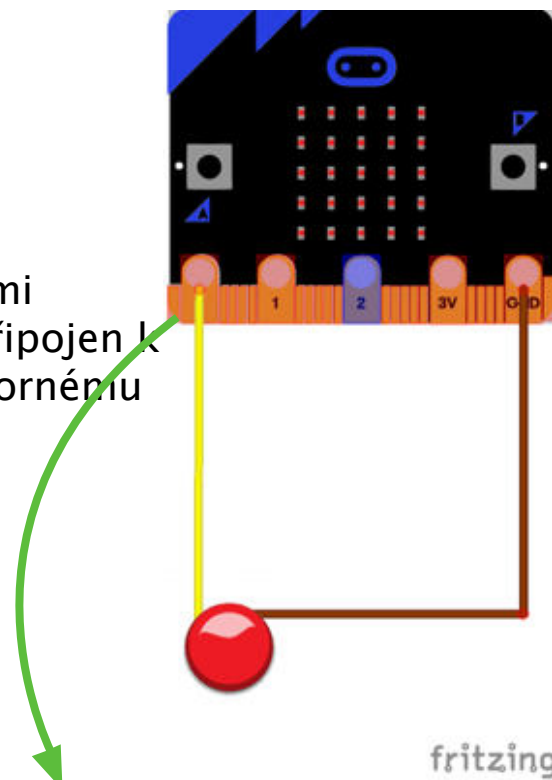
Předpoklad:

Mikrobity, které spolu komunikují, musí být ve stejné rádiové skupině (stejný kanál).

Hrát bzučák – Vysílač

Bzučák

Zapojení bzučáku (rádiový vysílač) je velmi jednoduché. Jeden kontakt bzučáku je připojen k pinu 0 – druhý kontakt je připojen k zápornému pólu (GND) mikrobitu.



V nestlačeném stavu je na pinu 0 mikrobitu 3V (stav 1). Stisknutím bzučáku se připojí pin 0 k zápornému pólu (GND) pomocí tlačítka a na pinu je 0V.



Stav pinu 0 je dotazován v kódovací části. Jakmile je toto stisknuto, text „Peter“ je odeslán rádiem. Pro každý z použitých bzučáků odešli jméno příslušného hráče (hráčů).

Hrát Bzučák – Vysílač

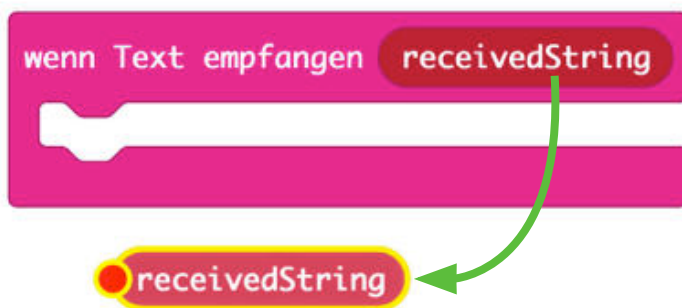
Bzučák (příjemce)



Pro zadání jmen potřebuješ z kategorií „Pokročilé“ a tento blok  `Text` .

Jakmile je text přijat, automaticky se uloží do vyrovnávací paměti „receivedString“ a může být dotazován jako konvenční proměnná.

Tip



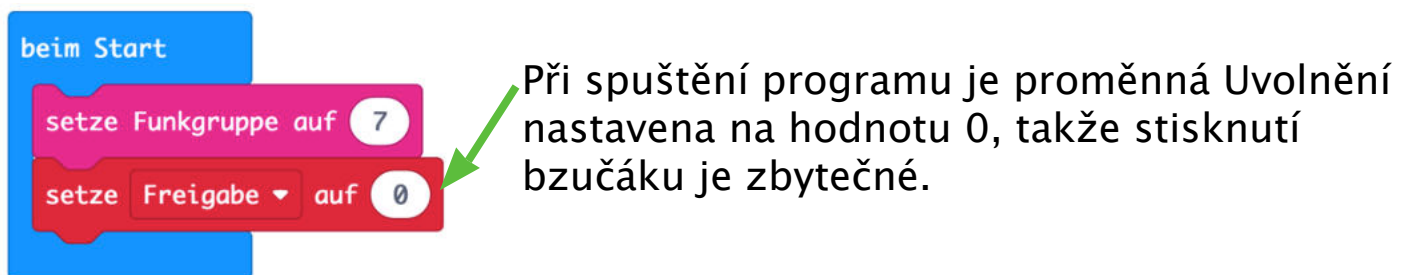
Pro použití proměnné „ReceivedString“ ji jednoduše přetáhneš z bloku „při přijímání textu....“ ven.

Hrát Bzučák – pokročilí

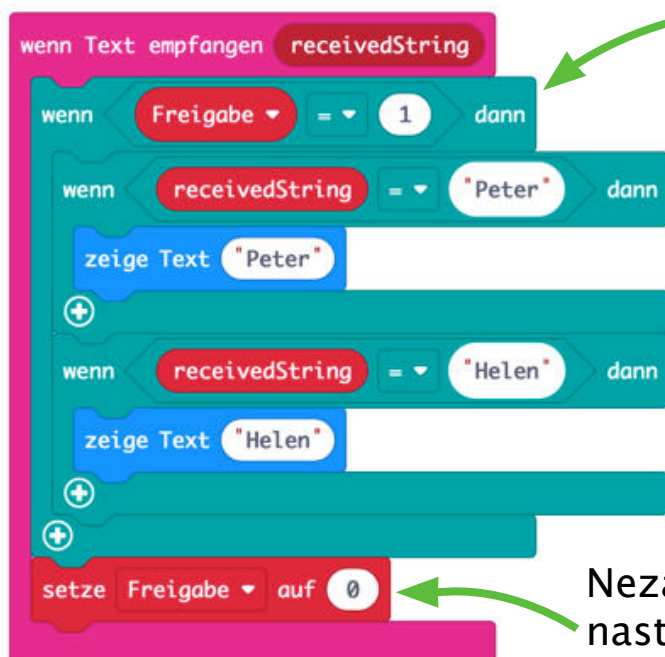
Problém

Rádiový přijímač zobrazuje sice název, jehož bzučák byl stisknut jako první, ale všechna další stisknutí bzučáku jsou uložena ve vyrovnávací paměti. Výsledkem by bylo, že by se jedno po druhém zobrazovala také všechna ostatní jména.

Tomu je třeba zabránit pomocí další proměnné (uvolnění). Stisknutí bzučáku se vyhodnocují pouze v případě, že proměnná „uvolnění“ je nastavena na hodnotu 1.



Kliknutím na tlačítko B se proměnná „uvolnění“ nastaví na 1, takže lze vyhodnotit další stisknutí bzučáku.



Dokud je proměnná uvolnění 1, vyhodnocuje se přijímaný rádiový text.

Nezapomeň na konci vyhodnocení nastavit proměnnou „uvolnění“ zpět na 0, aby se již nevyhodnocovaly stisknutí bzučáku (teprve po novém resetování tlačítkem B).

LCD displej bzučáku

Předchozí příklad rozšíříme tak, že LC displej u rozhodce ukáže jméno hráče, jehož bzučák byl stisknut jako první.



Předpokladem pro to je implementace LCD na rozhraní Makecode. Můžeš to provést vyhledáním výrazu „LCD“ v kategoriích „Pokročilí“ a „Rozšíření“.



Z navrhovaných možností zvolíš následující LC displej: **i2CLCD1602**.

Poté je k dispozici následující kategorie. Nejdůležitější bloky pro použití LC displeje:

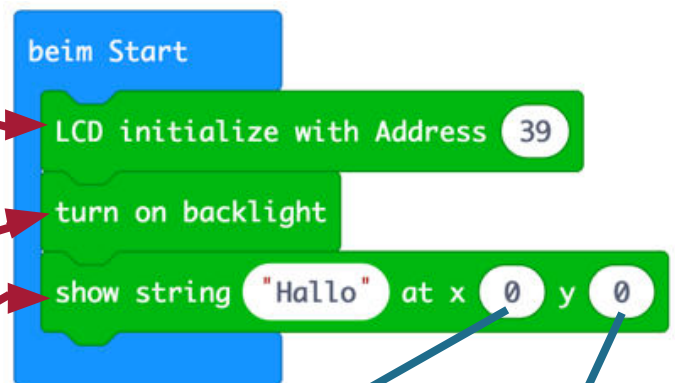


Aby bylo možné displej použít, musí být inicializován tímto řádkem.

Aby bylo možné displej použít, musí být inicializován tímto řádkem.

Podsvícení můžeš zapnout tímto způsobem.

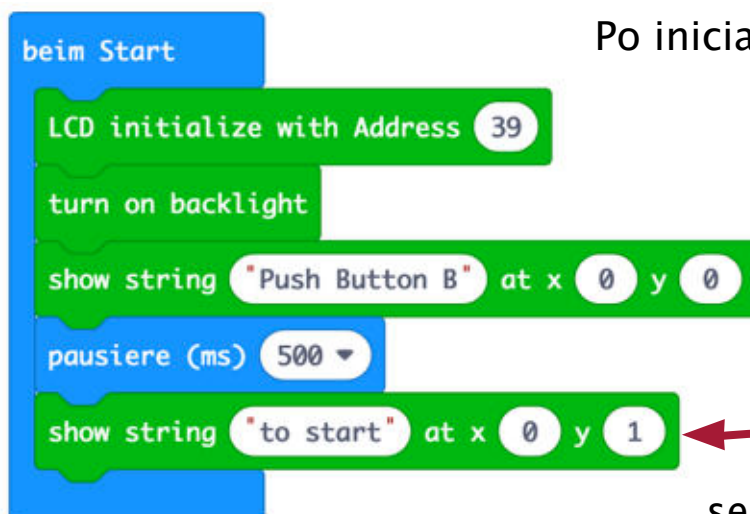
Pomocí tohoto řádku určíš, co by se mělo zobrazit na displeji.



1. značka

1.řádek

LCD Displej bzučáku – kódování



Po inicializaci displeje a zapnutí podsvícení

se v řádku 1 zobrazí „Stiskni tlačítko B“.

Stisknutím tlačítka B se vymaže displej a jsi připraven na druhé kolo.



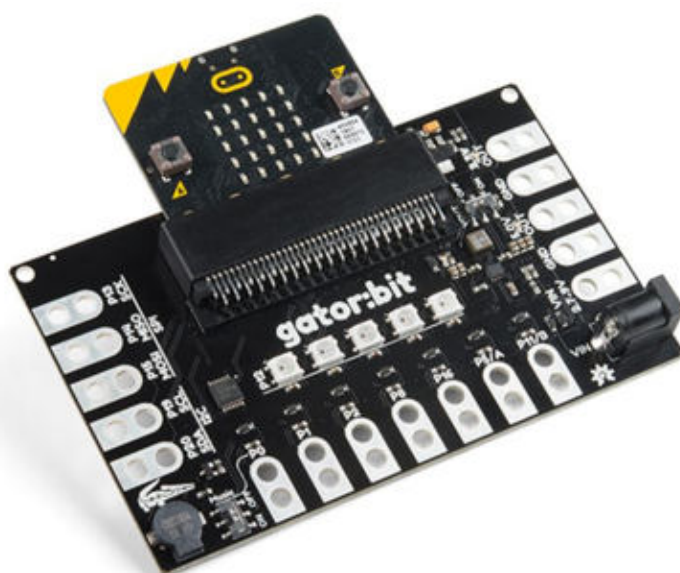
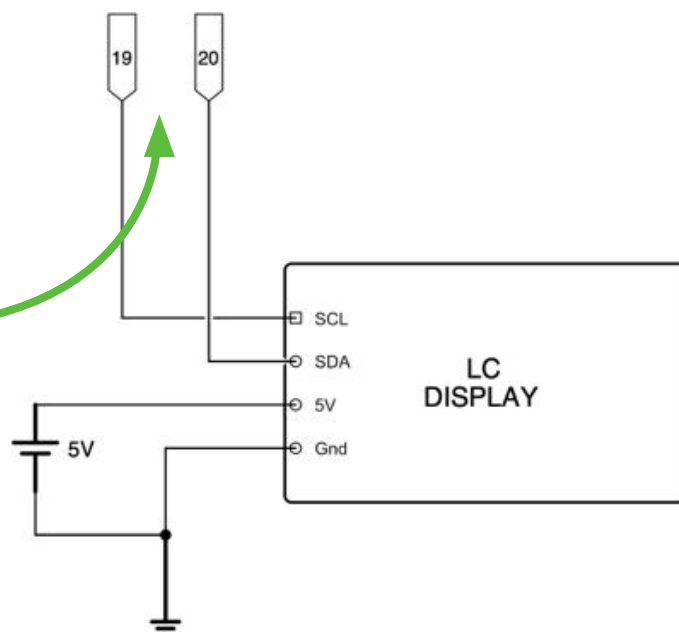
LCD Displej bzučáku – zapojení

Schéma zapojení vpravo ukazuje, jak zapojit LC displej k mikrobitu.

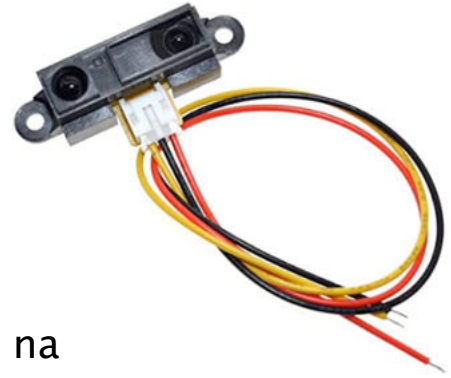
Připoj SCL k Pin19 a SDA k PIN20.

Důležité!

Potřebuješ externí 5V zdroj napájení, aby LC displej mohl spolehlivě fungovat. Může se jednat o powerbanku, externí napájení nebo, jako v našem případě, přídatnou desku pro mikrobit, která kromě zdroje napájení 5V také umožňuje odpojit nejdůležitější kolíky pomocí krokodýlí svorky.



IR snímač vzdálenosti – teorie



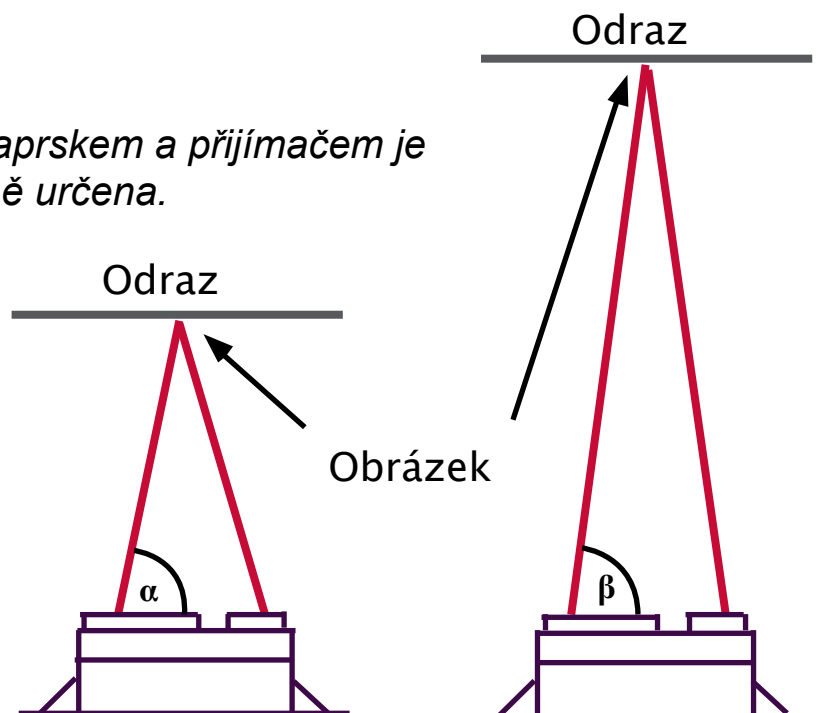
Teorie

Infračervený snímač vzdálenosti Sharp vydává na jedné straně infračervený paprsek.

Pokud paprsek nenarazí na překážku, neodráží se a přijímač na druhé straně neobdrží žádné informace.

Pokud se IR paprsek odrazí od překážky, zasáhne přijímač v určitém úhlu (v závislosti na vzdálenosti od překážky) a vzdálenost se vypočítá interně.

Pomocí úhlu mezi odraženým paprskem a přijímačem je vzdálenost mezi objektem interně určena.



Možnosti využití

- Parkovací asistent
- Robot
- Vysoušeč rukou
- Poplašné zařízení
- Splachování WC
- Otvírání dveří

IR snímač vzdálenosti – kódování

Zapojení

Dva napájecí kabely IR snímače můžeš spojit přímo s mikrobitem. [červená -> 3V/černá -> GND]. Připojíš žlutý kabel k pin 0,1 nebo 2.

V kategorii  Pins najdeš blok  analogické hodnoty pinu P0.

Tato hodnota dává hodnotu od 0 do 1023, v závislosti na vzdálenosti. Čím vyšší je hodnota, tím kratší je vzdálenost k překážce.



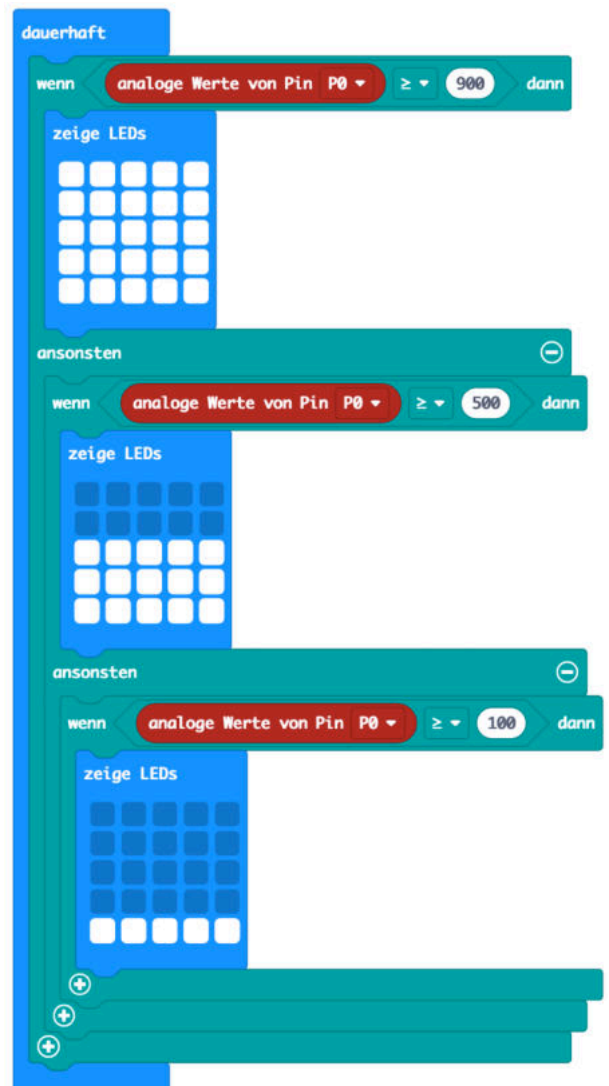
Se srovnávacím operátorem  z kategorie Logika můžeš prověřit, zda je určitá hodnota překročena, a podle toho provést pokyny.

Parkovací

Pro parkování přesné na centimetr, bylo by užitečné získat vizuální zpětnou vazbu na displeji.

Jediné, co pro to musíš udělat, je dotázat se na tuto hodnotu pin0, ke kterému je IR senzor připojen.

Čím vyšší je hodnota, tím více pruhů (kratší vzdálenost) se zobrazí.

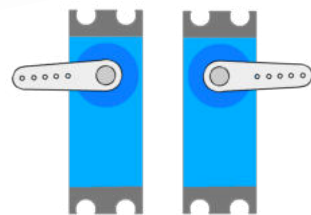


Servo – obecné

Fungování

Na rozdíl od běžného motoru se servo motor neotáčí o 360° , ale zaujímá určitou polohu v rozmezí 0° až 180° . Obyčejné servo se může otáčet jen mechanicky v rozsahu 0° až 180° , protože má zarážku (výjimka: 360° Continuous Rotation Servo).

Kromě napájecího zdroje (hnědá – 0V, červená – 5V) je k dispozici ovládací kabel (žlutý). Doba trvání impulsu na této lince určuje úhel serva. Každých 20 ms (0,02s) očekává servo impuls, který určuje úhel od 1ms [0°] do 1ms [180°].



Info

Servo SG90 lze provozovat přímo na mikrobitu se sníženým výkonem. Nejjednodušší způsob použití více serv je s vyhrazenou servo deskou nebo externím zdrojem napájení. Servomotory mají širokou škálu použití. Používají se v průmyslu, ve strojírenství, ale také v rámci různých hobby:

Možnosti použití

Průmysl:

Volnočasová oblast:

Motorová vozidla:

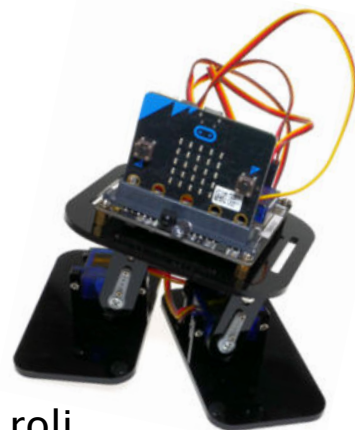
Senzorika:

Robotické rameno

Modelování

automatické nastavení sedadla

polohování senzorů

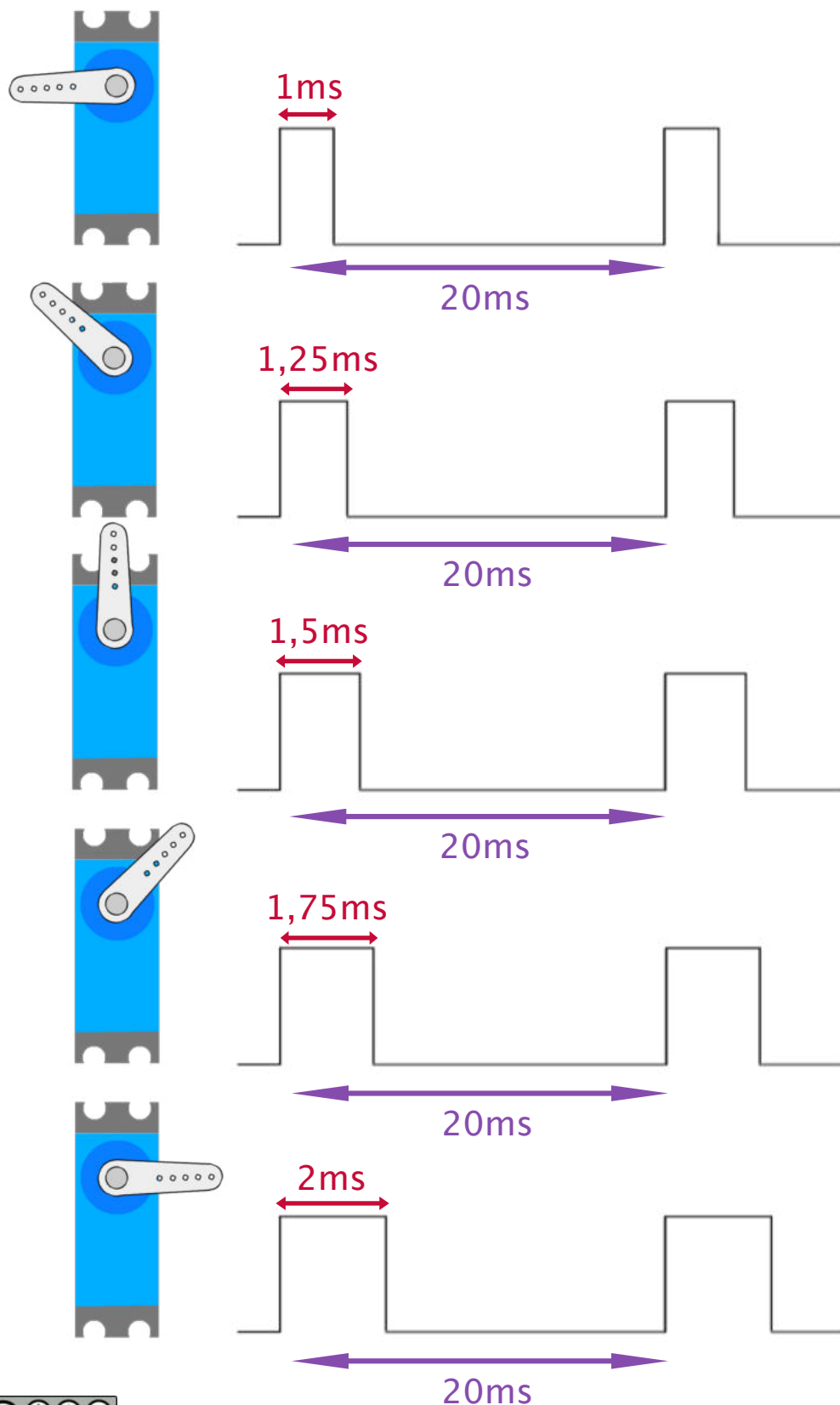


Servomotory se často používají, když hrají významnou roli vysoké točivé momenty a přesné, rychlé pohyby

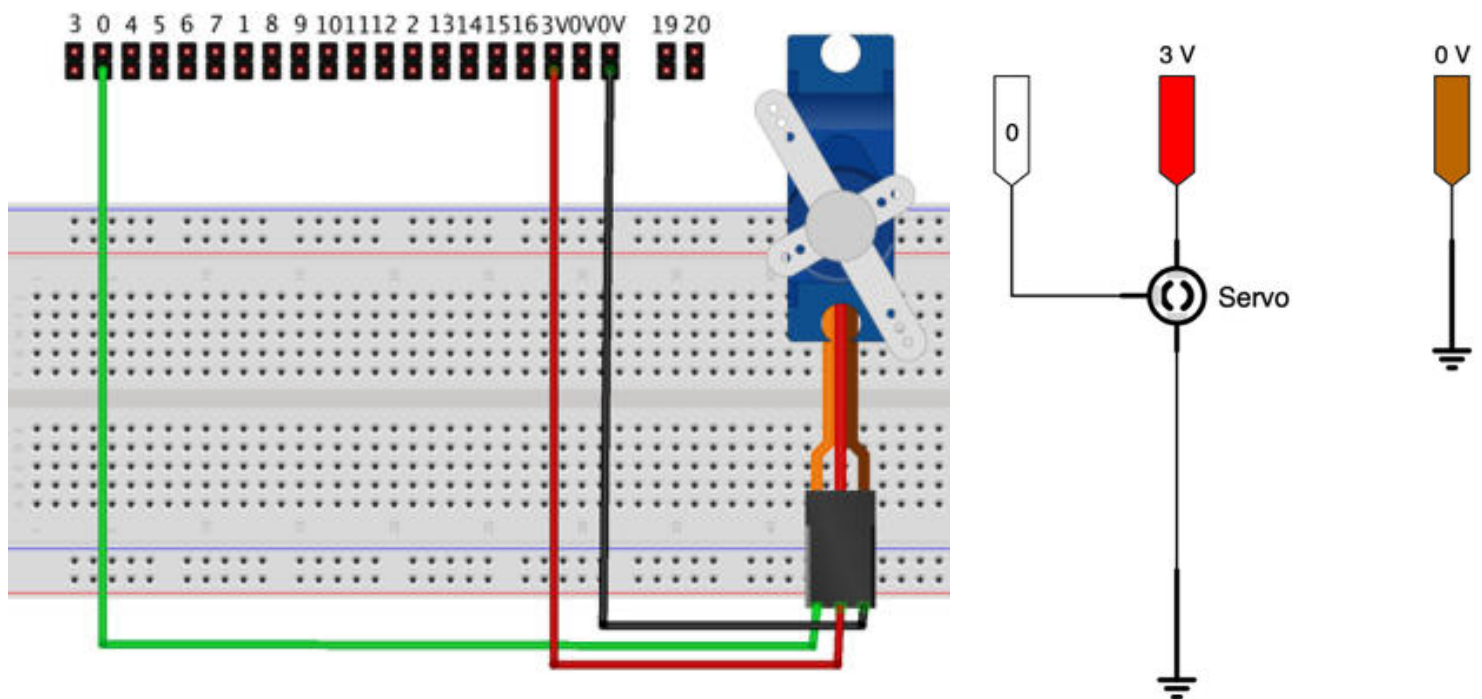
Servo – teorie

Během 20 ms indikuje šířka impulsu řídicího signálu úhlovou polohu serva. Při šířce impulsu 1ms přebírá servo úhel 0° (zcela vlevo) a pohybuje se se zvyšováním na 2 ms na 180° (úplně vpravo). Tyto hodnoty jsou pouze orientační a musí se převzít z datového listu.

Načasování provádí jednočipový počítač (mikrobit) a je implementováno během programování tzv. modulací šířky impulsu.



Servo – zapojení



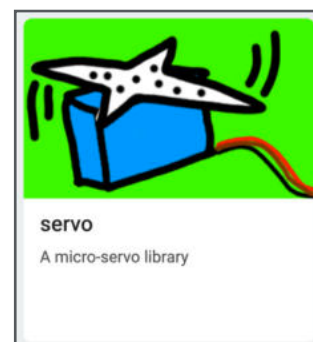
- Servo zapojení: pomocí **zeleného kabelu** vedeš připojení zcela vlevo (ovládací kabel) k **Pin0** mikrobitu.
- Střední konektor je připojen k 3V pinu pomocí **červeného kabelu** k **3V pinu**.
- Nakonec je Gnd serva spojeno s 0V pinem mikrobitu pomocí **černého kabelu**.

Info

Ve většině případů servo vyžaduje více proudu, než může mikrobit poskytnout. Pokud je tomu tak (servo se nepohybuje nebo se „třeše“), musíš servu dodat vlastní zdroj napájení. Přitom nezapomeň připojit vnější materiál (záporný pól) k pinu 0V mikrobitu.

Servo – kódování

Předpokladem pro to je implementace serva na rozhraní Makecode. Můžeš to provést vyhledáním výrazu „Servo“ v kategoriích „Pokročilí“ a „Rozšíření“.



Poté je k dispozici následující kategorie:

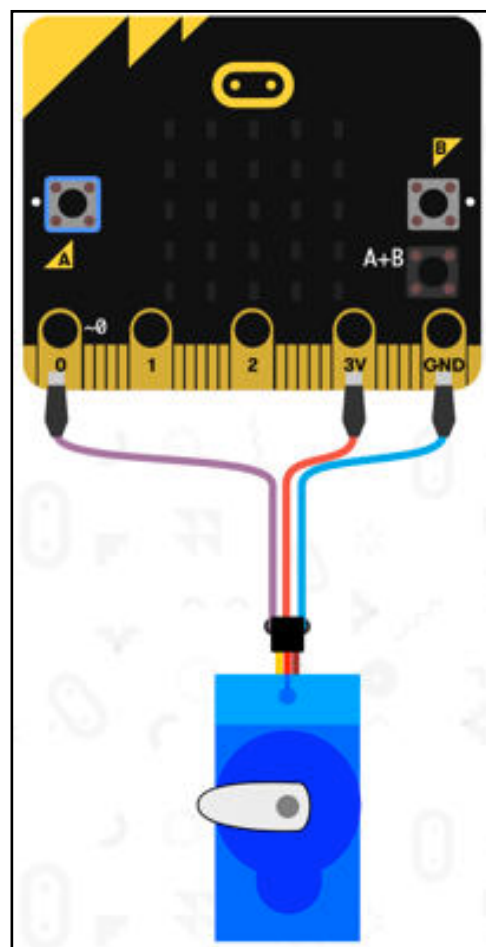


Otestuj funkci serva přiřazením 3 úhlových poloh tlačítkům.



Info

Funkci můžeš také skvěle otestovat bez hardwaru pomocí online simulátoru v rozhraní Makecode.



Servo – jas displeje

V tomto příkladu se okolní jas zobrazí s ukazatelem umístěným na servu.

K tomu potřebuješ blok **Lichtstärke** a dále ten blok, který používáš k určení úhlu serva.

setze Winkel von Servo an Servo P0 auf 0°

Problém

dauerhaft
setze Winkel von Servo an Servo P0 auf Lichtstärke°

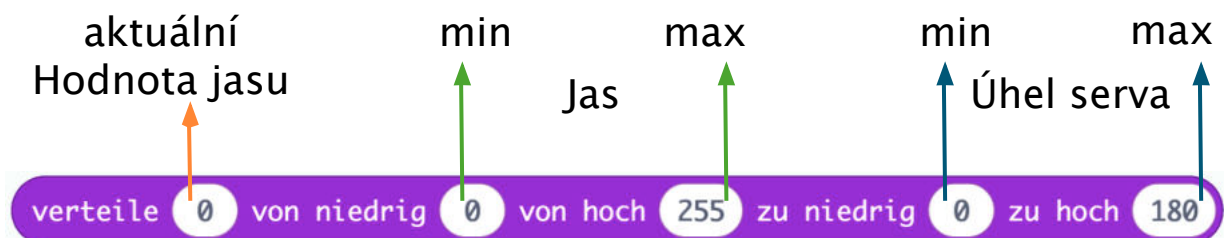
Pro vysílání intenzity světla jako úhel serva, viz příklad výše, by se rozsah hodnot neshodoval.

Rozsah hodnot serva je 0–180 (stupňů), zatímco rozsah hodnot snímače jasu je 0–255 v simulátoru a 0–1023 na samotném mikrobitu.

Řešení

Pro zobrazení číselné hodnoty z jedné oblasti do druhé, můžeš použít takzvanou funkci „rozdělit“ z kategorie „matematika“.

V našem případě by měly být možné hodnoty jasu 0–255 simulátoru přiřazeny možným hodnotám serva 0–180.



Takto vypadá hotový kód.

dauerhaft
setze Winkel von Servo an Servo P0 auf verteile Lichtstärke von niedrig 0 von hoch 255 zu niedrig 0 zu hoch 180°

Důležité: Pokud používáš „skutečný“ mikrobit a ne online simulátor, musíš nastavit max. hodnotu jasu od 255 do 1023.

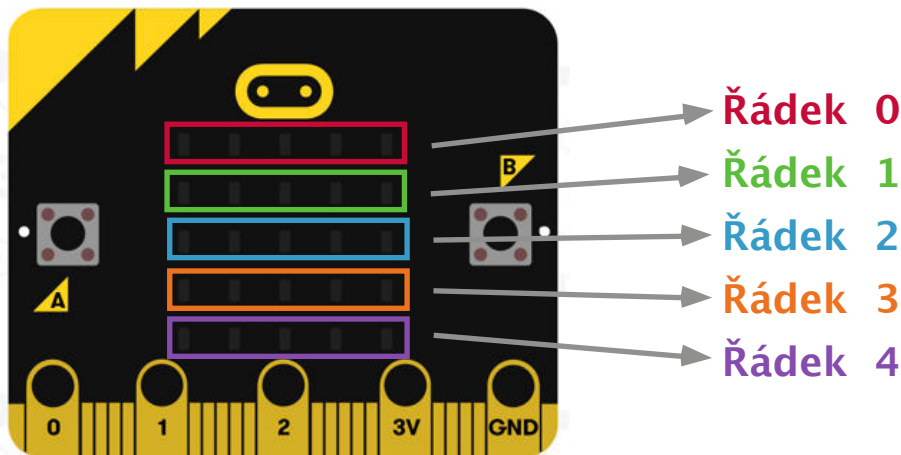
Kreslení s joystickem – 1

Pohybuj pomocí joysticku LED pixel na mikrobitu. Při každém stisknutí tlačítka (tlačítko A) se uloží aktuální pozice pixelu. Stisknutím tlačítka B se zobrazí celkový obraz všech uložených pixelů. To můžou být ikony, emotikony, písmena, nebo jednoduché obrázky.

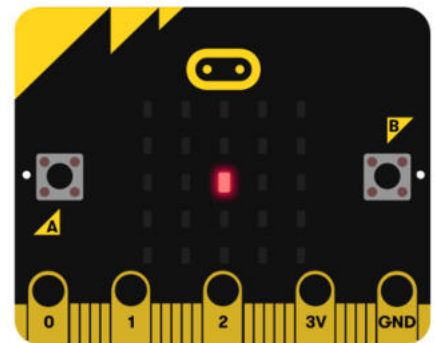
K tomu potřebuješ blok z kategorie LED.

Zeichne x 0 y 0

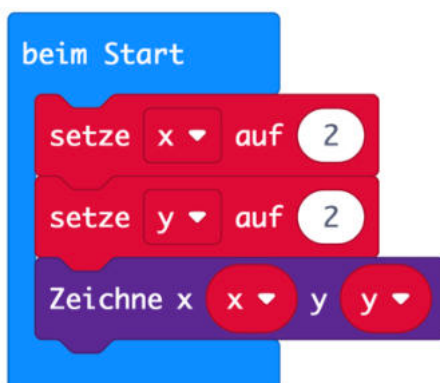
Stejně jako v souřadnicovém systému znamená x sloupce a y řádky. Pro programovací jazyky je obvyklé začít počítat od 0. První řádek je tedy řádek 0 a poslední řádek je řádek 4.



Pokud chceš, aby se pixel objevil na začátku uprostřed, lze toho dosáhnout pomocí souřadnic $x=2$ a $y=2$.



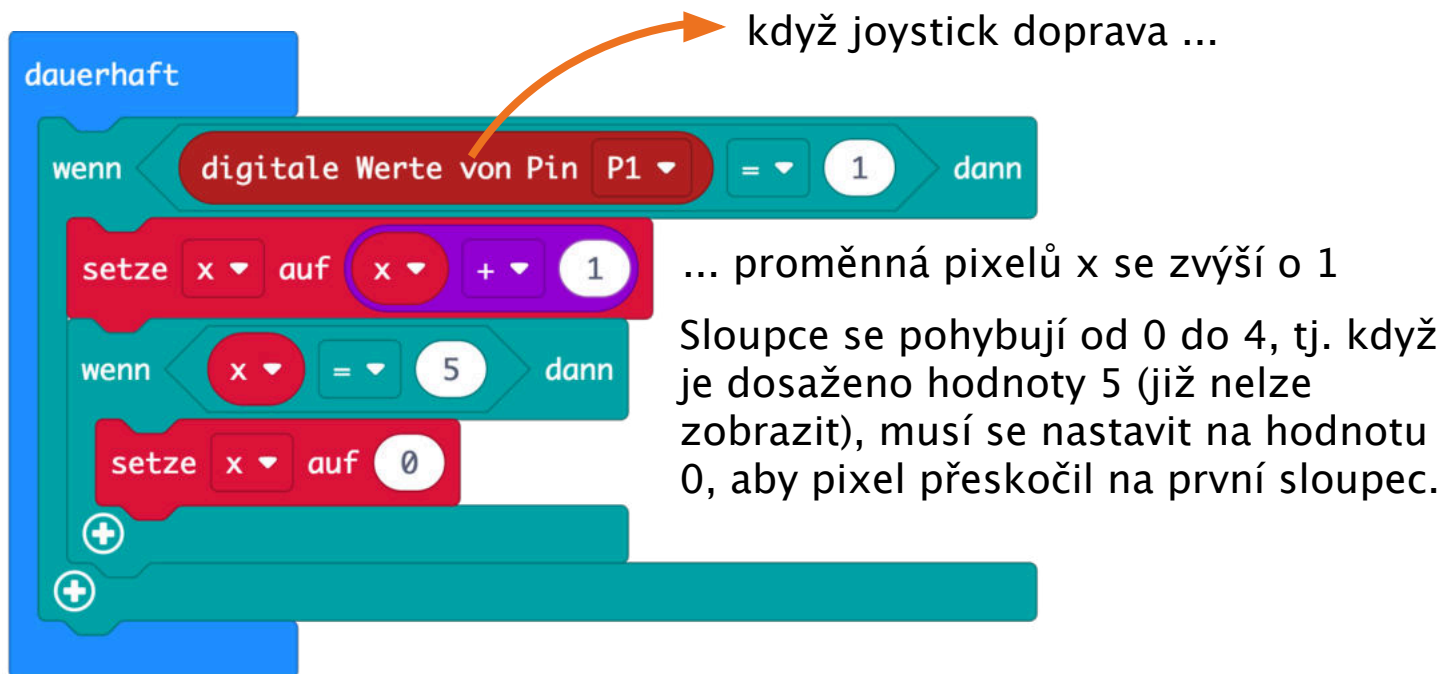
Vzhledem k tomu, že zářící pixel by měl být „pohyblivý“ pomocí joysticku, nesmí být statický, nýbrž dynamický, tj. specifikován proměnnými.



Výstup na displeji je stejný – ale odteď může být nastav x na 2 hodnota LED zvýšena nebo snížena pomocí nastav y na 2 joysticku tak, aby se LED pixel pohyboval.

Kreslení s joystickem – 2

Následující programovací blok umožňuje, aby se LED pixel pohyboval doprava, jakmile je joystick stisknut doprava (pin 1) (za předpokladu, že bylo provedeno příslušné zapojení).



Zapojení:

Joystick není nic víc než 4 tlačítka, z kterých každé uzavře obvod, když je odpovídajícím způsobem obsluhováno.

Vzhledem k tomu, že mikrobit na digitálním vstupu rozpozná napětí 0 voltů jako „stisknuto“, spojí se jeden kabel s GND (0 voltů) a druhý je k příslušnému pinu mikrobitu, například Pin1. V okamžiku stisknutí tlačítka (např. joystick vpravo) je obvod uzavřen a na Pin1 je 0 voltů, které jsou mikrobitem rozpoznány jako stisknutí klávesy.

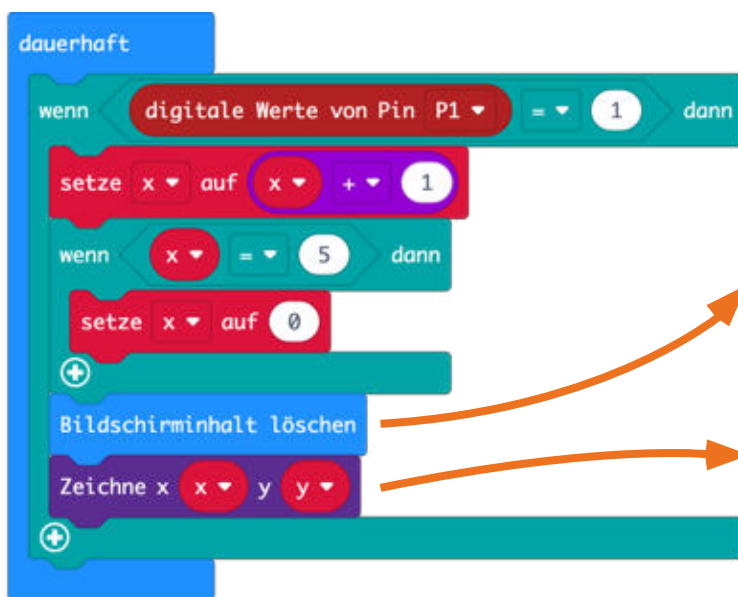
Kreslení s joystickem – 3

Chceš-li zobrazit také pixel (v tuto chvíli byla upravena pouze odpovídající proměnná), musí být přidán následující blok.



Jelikož se při stisknutí joysticku doprava neodstraní existující pixel, při každém tisku by byl přidán další pixel.

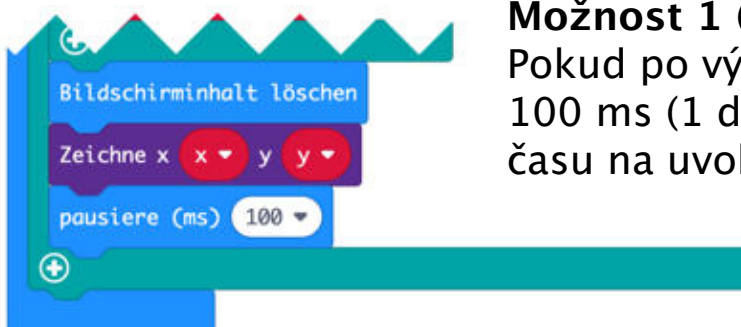
Chceš-li odstranit předchozí pixel, musí být obrazovka před výstupem aktuálního pixelu vymazána.



... proměnná pixelů x se zvýší

Vymazání předchozího

Vzhledem k tomu, že mikrobit provádí pokyny závratnou rychlostí, i rychlé stisknutí tlačítka by okamžitě posunulo pixel o mnoho míst dále. Existují 2 způsoby, jak tomu zabránit:



Možnost 1 (kompromis):

Pokud po výstupu pixelu zakomponujete pauzu 100 ms (1 desetina sekundy), máte dostatek času na uvolnění tlačítka po jeho stisknutí.

Upozornění

Pokud stisknete tlačítko nebo joystick příliš dlouho (více než 100 ms), bude se pixel stále pohybovat. Pokud bude pauza příliš dlouhá, nebudeš moci posovat pixel dostatečně rychle.

Kreslení s joystickem – 4

Možnost 2 (elegantní metoda):

Po stisknutí tlačítka → čeká se pomocí prázdné smyčky, dokud se tlačítko (Pin1) znovu neuvolní.



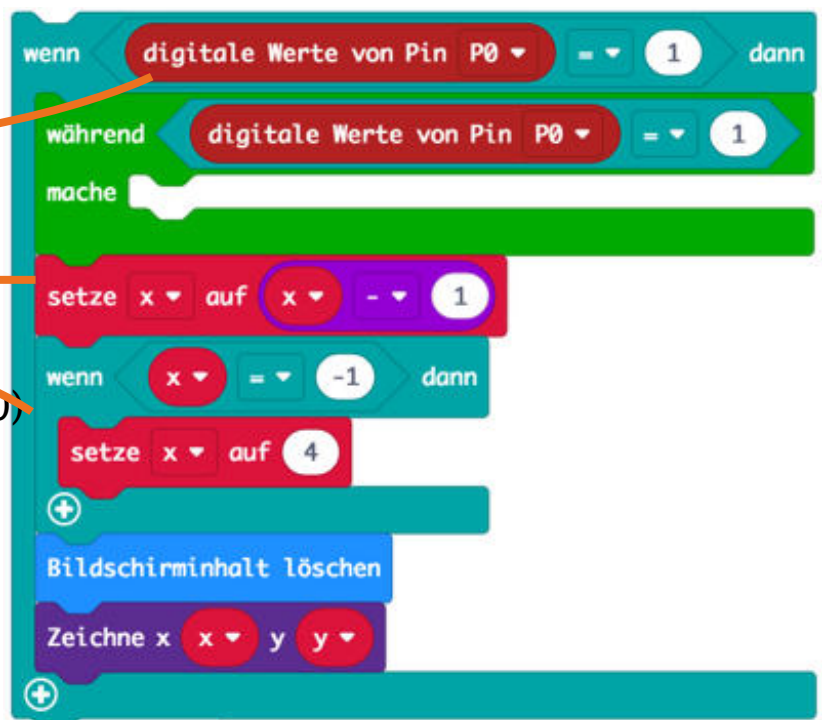
Program zůstává v této „čekací smyčce“, pokud je stisknuto tlačítko. Jakmile je tlačítko uvolněno, kód programu pokračuje pod ním. Nezáleží tedy na tom, jak dlouho mačkáš tlačítko – interpretuje se to pouze jako jedno stisknutí tlačítka.

Po stisknutí tlačítka → čeká se pomocí prázdné smyčky, dokud se tlačítko (Pin1) znovu neuvolní.

... proměnná pixelů x se zvýší o 1

Hodnota pixelu se sníží o 1.

Pokud je sloupec zcela vlevo (x=0) snížen o 1 (x=-1), musí být nastaven na x=4 zcela vpravo.



Celkové ovládání pixelu pomocí joysticku:

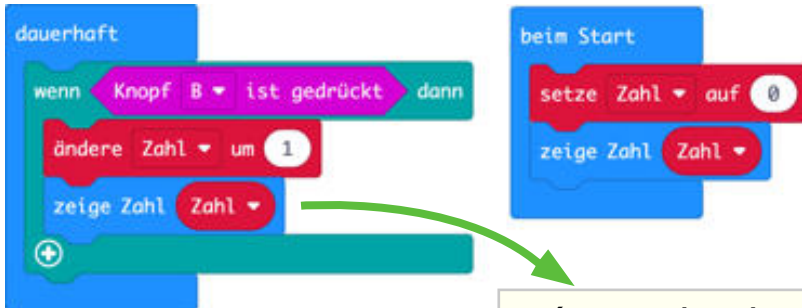
https://makecode.microbit.org/_0VxWRVh5rhLe

Stisknutí klávesy pro pokročilé – problém

Úkol:

Při startu se na displeji zobrazí číslo 0.

Pokaždé, když stiskneš tlačítko B, měl by být zvýšen výstup o 1.



Výstup displeje je standardně nastaven na 0,6 sekundy (600 ms).

Upozornění

Dobu trvání zobrazení můžeš změnit přidáním čárky za výstupní hodnotu v JavaScriptu nebo Pythonu a zadáním trvání v milisekundách (1 sekunda = 1000ms).

```
1 Zahl = 0
2 basic.show_number(Zahl)
3
4 def on_forever():
5     global Zahl
6     if input.button_is_pressed(Button.B):
7         Zahl += 1
8         basic.show_number(Zahl, 200)
9 basic.forever(on_forever)
10
```

Problém

Pokud je výstupní čas příliš krátký, zobrazení vyskočí dvakrát nebo vícekrát při jednom stisknutí klávesy.

Pokud je výstupní čas příliš dlouhý, musíš počkat dlouho, dokud nebude rozpoznáno další stisknutí klávesy nebo stisknutí klávesy nebude rozpoznáno vůbec.

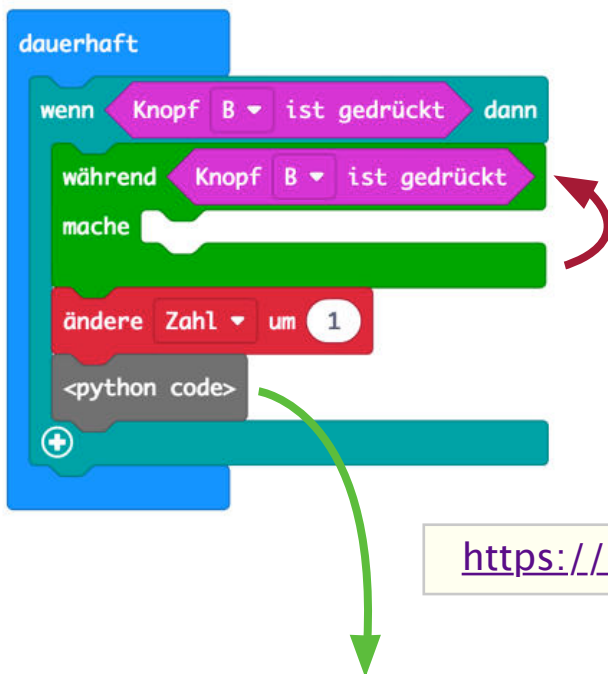
Stisknutí klávesy pro pokročilé – řešení

Řešení problému:

Elegantním řešením by bylo vyhodnotit stisknutí klávesy až po uvolnění tlačítka. Je tedy na uživateli, jak krátce nebo dlouze stiskne tlačítko, detekuje se to pouze jednou.

Implementace

Po stisknutí tlačítka se čeká, až se tlačítko znovu uvolní. Teprve potom je příkaz proveden.



Program pokračuje až po uvolnění tlačítka.

Vzhledem k tomu, že se při čekání nic neděje, blok „make“ zůstává ve smyčce prázdný.

https://makecode.microbit.org/_9bpi07HUJasw

Aby bylo rozpoznáno i krátké stisknutí tlačítka, zkrát dobu zobrazení na cca 10 ms.

```
basic.show_number(Zahl, 10)
```

Upozornění

Zobrazení víceciferných čísel by muselo být přizpůsobeno delšímu výstupnímu času na displeji pomocí samostatného dotazu, aby výstup nebyl příliš rychlý.

https://makecode.microbit.org/_Fsr7p915mh7c

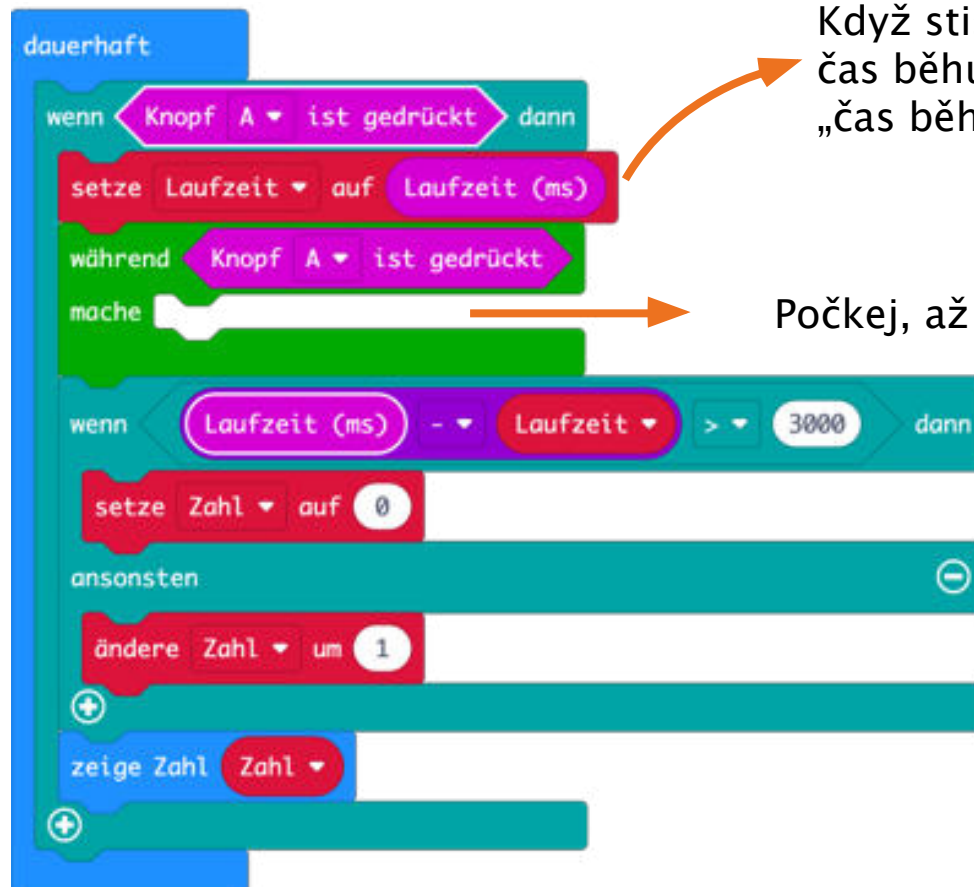
Rozdíl Krátké – dlouhé stisknutí tlačítka

Zadání :

Jediným tlačítkem by se mělo jeho krátkým stisknutím zobrazit další číslo. Displej by měl být resetován na 0 dlouhým stisknutím tlačítka. Tento scénář lze nalézt například u hardwaru, který nemá dostatek místa pro více tlačítek. Nebo se pokusit provést možný reset neúmyslně stisknutím tlačítka po dlouhou dobu.



Pro řešení tohoto scénáře je zapotřebí jakýsi „druh stopek“, který oznamuje, zda bylo tlačítko stisknuto dostatečně dlouho. Tento úkol přebírá tzv. časovač. Časovač se spustí automaticky, když je mikrobit napájen a počítá po mikrosekundách.



Když stiskneš tlačítko, aktuální čas běhu se uloží do proměnné „čas běhu“.

Počkej, až se tlačítko uvolní.

Pokud bylo stisknutí klávesy delší než 3 sekundy, číslo se resetuje na 0.

https://makecode.microbit.org/_8MWLTCPR8fDx

Vylepšení Stisknutí tlačítka krátký – dlouhý

V předchozím příkladu byl displej resetován až po uvolnění tlačítka, když bylo tlačítko stisknuto delší dobu. Tento reset by měl být nyní zobrazen okamžitě, jakmile se překročí čas.



V této smyčce se nejen dotazuje, zda je tlačítko uvolněno, ale také, zda bylo stisknuto déle než 3 sekundy. Pokud tomu tak je -> displej se resetuje na 0.

https://makecode.microbit.org/_autMpUT7gJE7



Microbit – mapa pinů

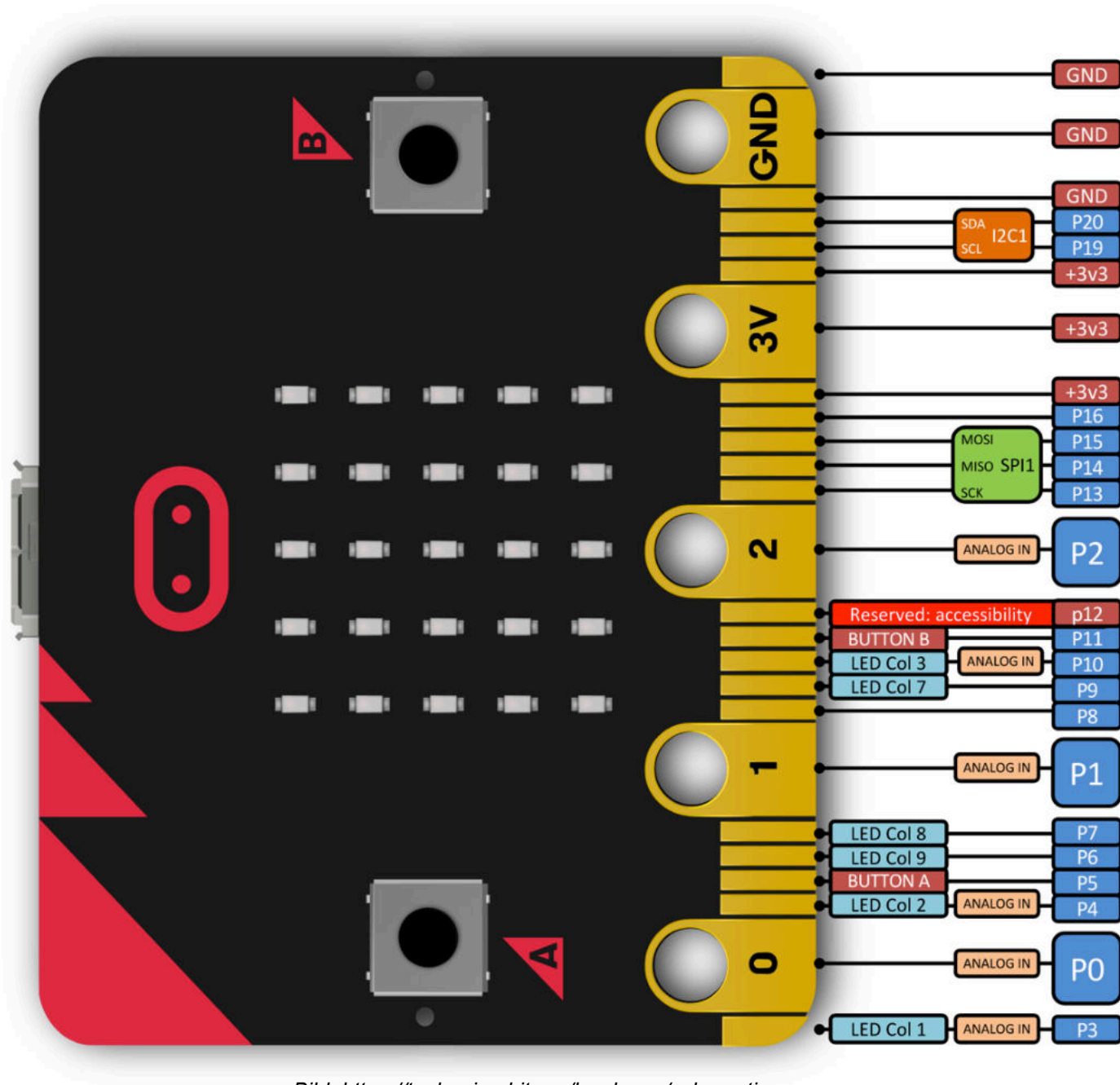


Bild: <https://tech.microbit.org/hardware/schematic>