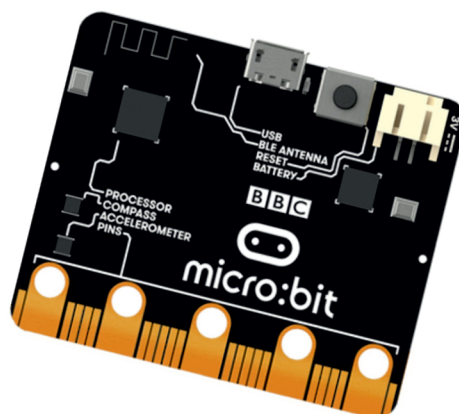
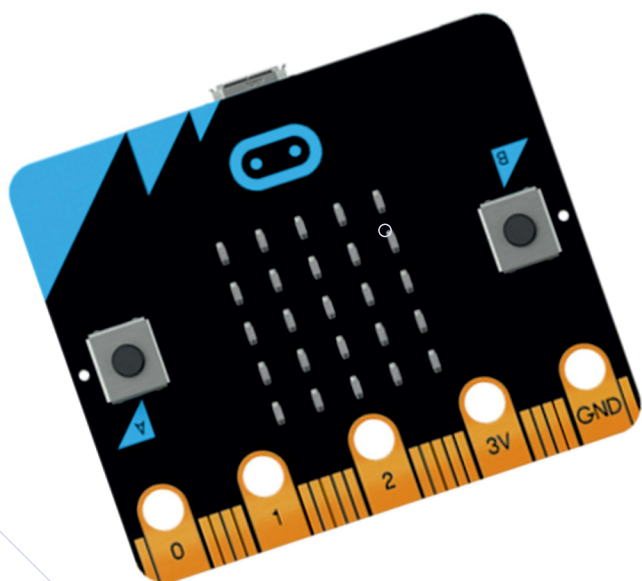


programovací manuál **MICRO:BIT**



MICRO: BIT

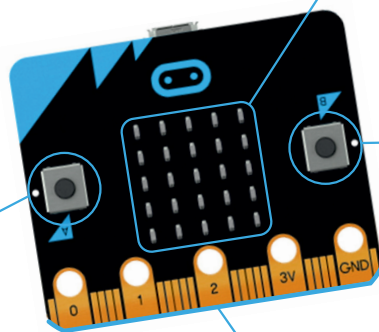
Micro:bit je mikropočítač navržený britskou společností BBC vytvořený pro podporu technického vzdělání. Díky němu můžete vytvořit množství elektronických a robotických zařízení. Je založen na 32 bitovém procesoru s jádrem ARM Cortex-M0. Komunikuje skrze rozhraní USB. To slouží k napájení i nahrání programu. K napájení slouží také bateriový konektor, popřípadě napájecí piny.

Destička je stavěna na napájení 3V.

5x5 LED Matice

zobrazení pro micro:bit obsahuje 25 namontovaných červených LED diod

Přední tlačítko A
integrované tlačítko



Přední tlačítko B
integrované tlačítko

VSTUP/VÝSTUP Připojení

tři vstupní/výstupní připojení pinů
a napájecí a uzemňující (3V a GND)

Mikro USB Konektor

používá se pro napájení a programování,
když je připojen k počítači

Bluetooth anténa

Nordic Semiconductors nRF51822

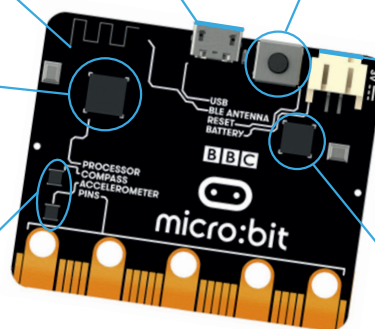
hlavní mozek desky poskytující
centrální procesorovou
jednotku, bluetooth
a paměťové funkce

Resetovací tlačítko programu

ovládání hard resetu, který restartuje desku
a program

Napájecí konektor

připojení pro poskytování
napájení prostřednictvím
baterií nebo jiného zdroje



Kompas a akcelerometr

integrované tři osé součástky pro
snímání směru a pohybu

Freescale Kinetis K126 Microcontroller (MCU)

ovládá USB konektor

Pro naprogramování vašeho micro:bitu je ideální použít Microsoft Block editor.

<https://makecode.microbit.org/>

V tomto webovém rozhraní si můžete podle svých programátorských dovedností vybrat z grafického programování v podobě bloků či v JavaScriptu. Pro začátečníky doporučujeme první ze způsobů. Ten je založen na skládání jednotlivých příkazů grafické podoby k sobě.



V nabídce je také možnost výuky vyšších programovacích jazyků Micropython a C++.

odkaz na webovou stránku micro:bit

vytvoření či otevření projektu

výběr mezi programovacím jazykem

micro:bit Projekty Sdílet Bloky JavaScript Microsoft

Hledat...

Základní
Vstup
Hudba
LED
Rádio
Smyčky
Logika
Proměnné
Matematika
Rozšířené

start/stop
restart

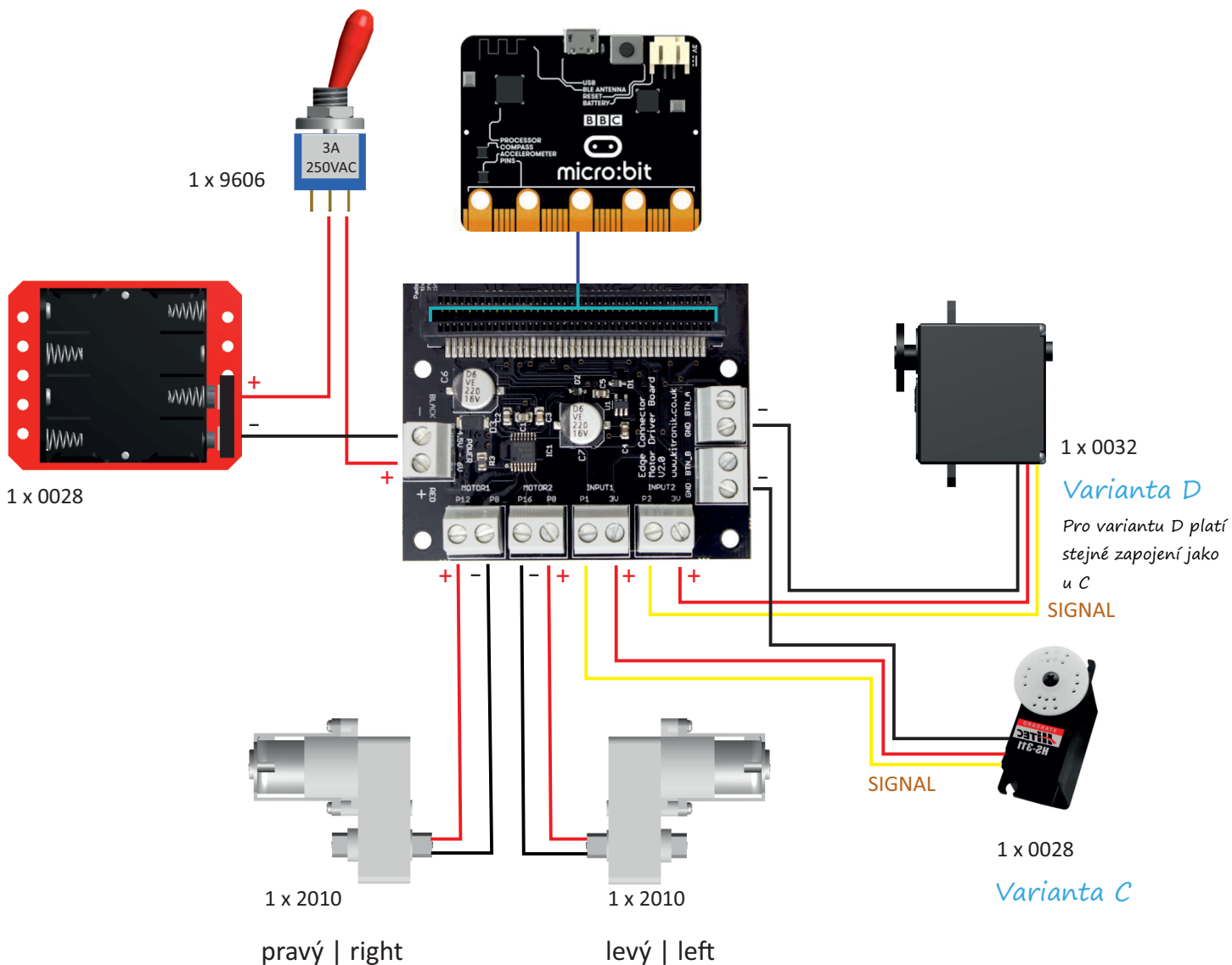
SIMULÁTOR MICRO:BITU

PŘÍKAZOVÁ LIŠTA

EDITAČNÍ OKNO PROGRAMU




Stáhnout Kitronik Line following bugg

nahrání programu na destičku
pojmenování a uložení programu
posun o krok vpřed nebo vzad v programu



MOTOR DRIVER BOARD V2.0

- 3V napájení (+)
z pravidla značený červenou barvou
- GND napájení (-)
z pravidla značený černou barvou
- BTN_A pin aktivuje tlačítko A
- BTN_B pin aktivuje tlačítko B

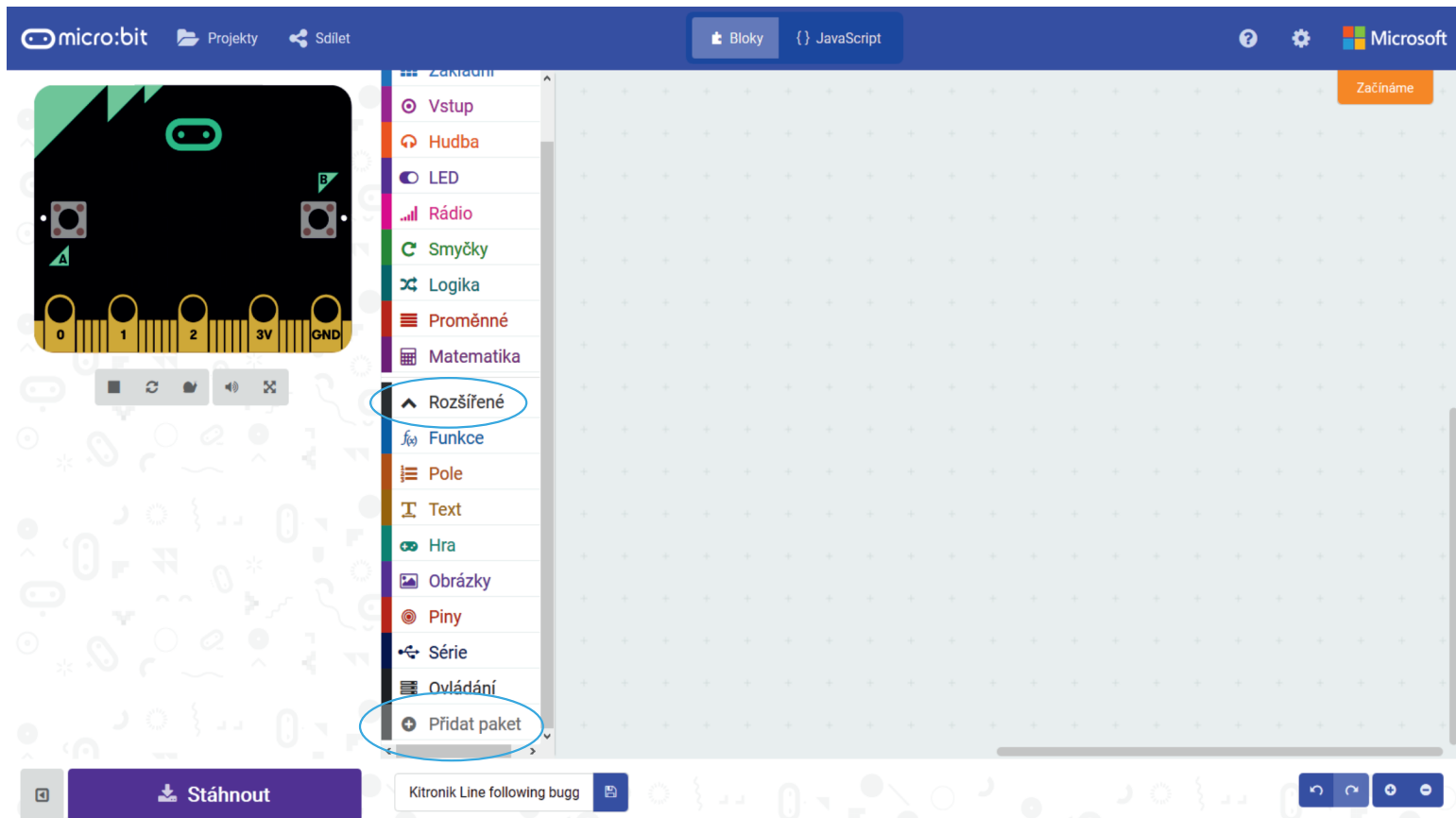
- PIN**  v překladu z angličtiny výraz kolík, špendlík
-  jedná se o vodivý „kolík“ je propojovací a slouží
-  k sestavování hardwaru

*Pokud máte již vše zapojeno,
tak můžeme přejít k programování.*

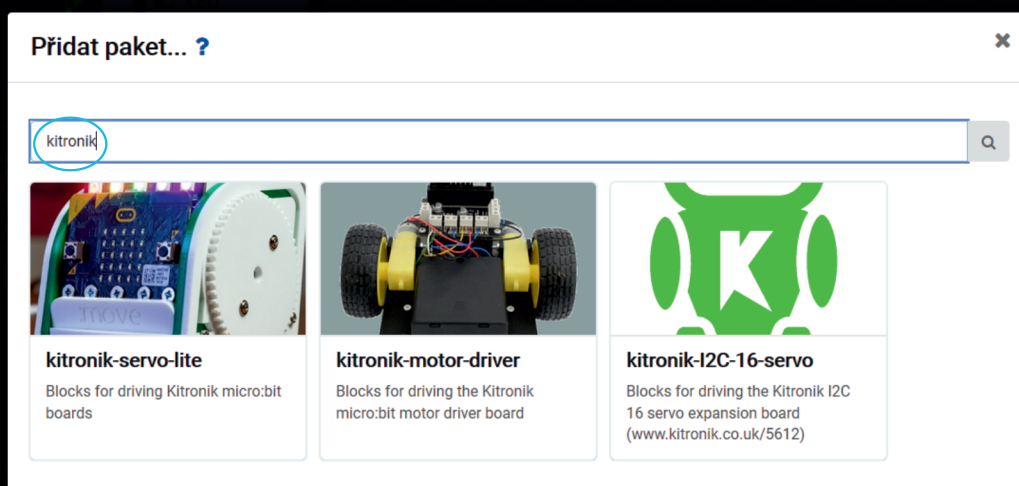
Otevřete si již zmíněné programovací prostředí pro micro:bit a můžeme začít. První si ověříme správné zapojení pomocí jednoduchého programu. Dále si vyzkoušíme naprogramovat robotovi trasu a pak ovládání vozítka micro: bitem. A nakonec vytvoříme program vhodný pro řízení mobilní aplikací kterou si můžete stáhnout. Ke všemu se postupně dostaneme.

Nyní si budeme muset do příkazové lišty nainstalovat balíček Kitronik pro řízení motorů. Najdeme v něm speciální příkazy pro jejich ovládání. A nyní, jak na to!

1. První rozbalíme nabídku Rozšířeně.
2. Dále úplně dole v nabídce vybereme možnost Přidat paket.

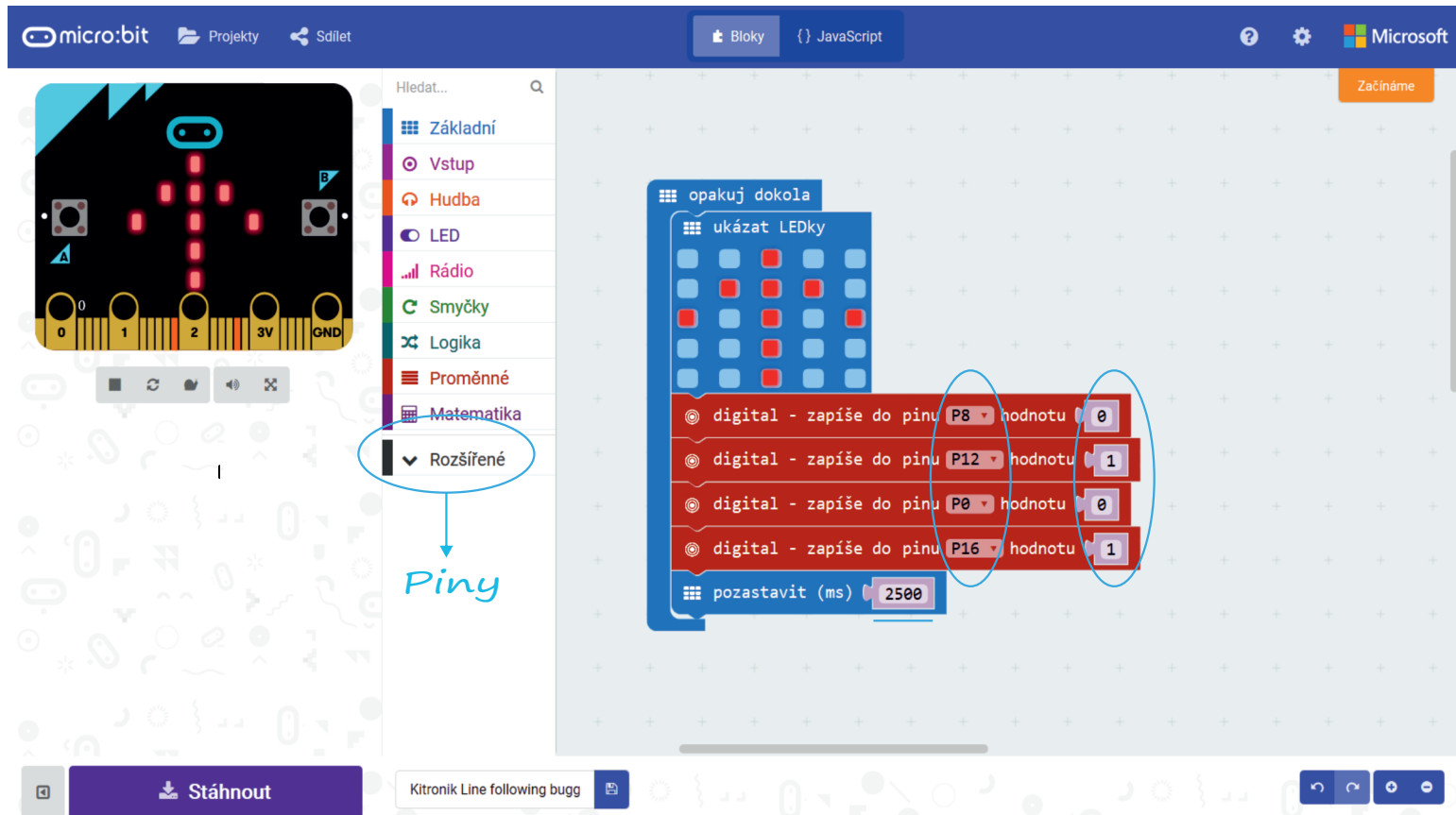


3. Nyní zadáme do vyhledávací lišty název balíčku - KITRONIK.
4. Z nabídnutých knihoven nainstalujeme kitronik-servo-lite a kitronik-motor-driver.

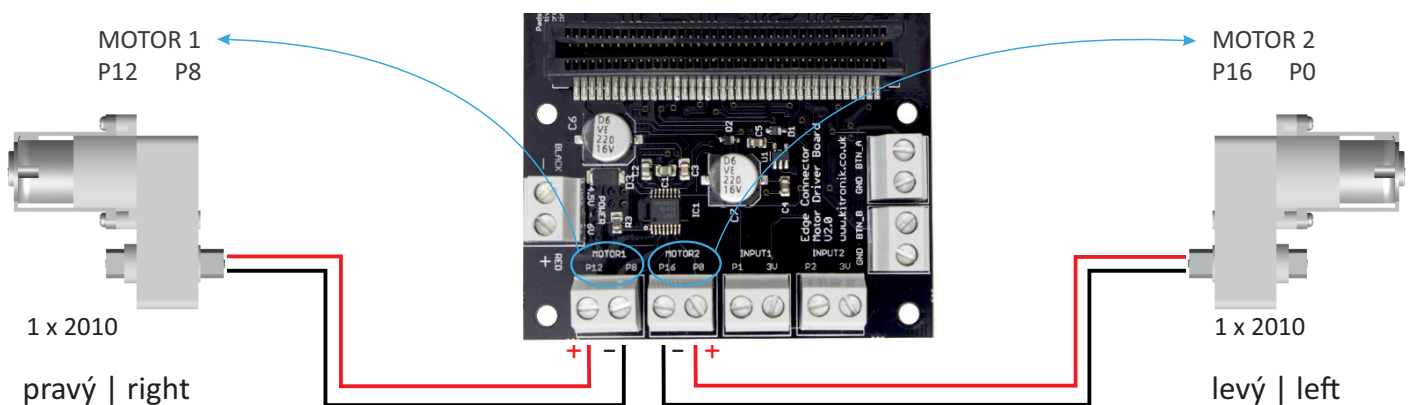


PRVNÍ PROGRAM který napíšeme bude námi vytvořená trasa vozítka nezávislá na ovládní. S parametry trasy si můžete libovolně hrát. Nastavením pozastavení určíte dobu jízdy do jednotlivých směrů.

3. Nyní si vybereme z příkazové lišty, z nabídek Základní a Piny příkazy které jsou zobrazené níže.
4. Díky příkazu opakuj dokola se vám bude program neustále opakovat. Bez něj se program vykoná pouze jednou.
5. Dále si zvolíme příkaz ukázat LEDky a zde si pro svou vlastní orientaci nakreslíme šipku určující směr jakým se vozítko rozjede. Ta se nám bude zobrazovat i přímo na micro:bit destičce. První nachystáme směr jízdy dopředu.
6. Dále vložíme příkazy vztahující se k nastavení pinů zobrazených níže.



7. Pro pin 8 nastavíme hodnotu 0 a pro pin 12 hodnotu 1. Dále na pin 0 nastavíme hodnotu 0 a na pin 16 hodnotu 1. Hodnoty 1 jsou tedy na opačných pólech (+ a -) a díky zrcadlovému otočení motorků se budou kolečka točit stejným směrem. V tomto případě dopředu.



8. Nyní si do programu přidáme příkaz pozastavit (ms). Zadaná hodnota nám určí délku času otáčení koleček (motorku) v ms. Krátká, ale na testování ideální zkušební délka je 2500 ms.

9. Další části programu jsou obdobné. Celý program naleznete níže. Nastavení pinů je pro každý směr různé. U směrů do boku nastavíme logickou 1 jen pro jeden pin ze čtyř. Ukázky směrů jízdy vidíte na obrázku pod textem.

The screenshot shows the micro:bit IDE interface. On the left is a virtual representation of the micro:bit board with a red line on it. The central pane shows a block-based program. The code is as follows:

```
opakuji dokola
  ukázat LEDky
  digital - zapiše doplnu P8 hodnotu 1
  digital - zapiše doplnu P12 hodnotu 1
  digital - zapiše doplnu P0 hodnotu 0
  digital - zapiše doplnu P16 hodnotu 1
  pozastavit (ms) 2500
  ukázat LEDky
  digital - zapiše doplnu P8 hodnotu 0
  digital - zapiše doplnu P12 hodnotu 1
  digital - zapiše doplnu P0 hodnotu 0
  digital - zapiše doplnu P16 hodnotu 0
  pozastavit (ms) 2500
  ukázat LEDky
```

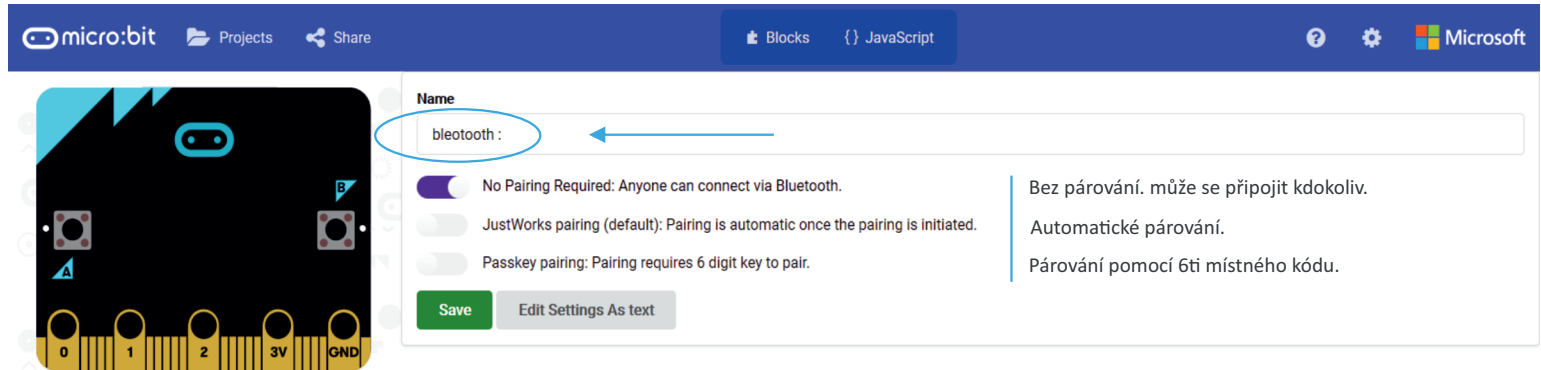
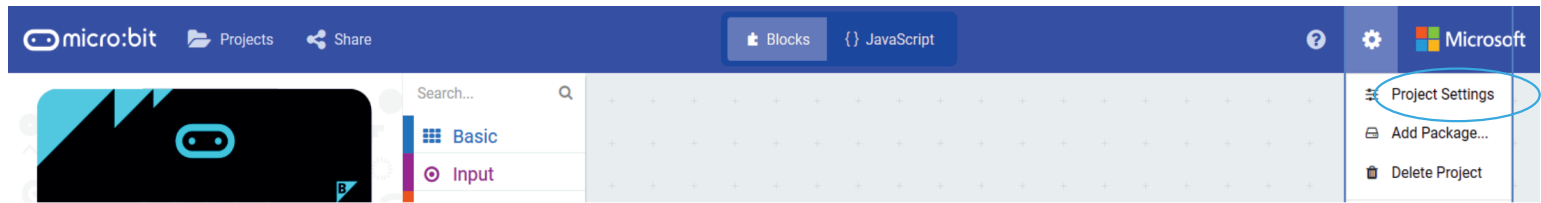
Red circles highlight the pin and value settings in the 'digital - zapiše doplnu' blocks.

The screenshot shows the micro:bit IDE interface. On the left is a virtual representation of the micro:bit board with a red line on it. The central pane shows a block-based program. The code is as follows:

```
digital - zapiše doplnu P8 hodnotu 0
digital - zapiše doplnu P16 hodnotu 0
pozastavit (ms) 2500
ukázat LEDky
digital - zapiše doplnu P8 hodnotu 0
digital - zapiše doplnu P12 hodnotu 0
digital - zapiše doplnu P0 hodnotu 1
digital - zapiše doplnu P16 hodnotu 1
pozastavit (ms) 2500
ukázat LEDky
digital - zapiše doplnu P8 hodnotu 1
digital - zapiše doplnu P12 hodnotu 0
digital - zapiše doplnu P0 hodnotu 1
digital - zapiše doplnu P16 hodnotu 0
pozastavit (ms) 2500
```

Red circles highlight the pin and value settings in the 'digital - zapiše doplnu' blocks.

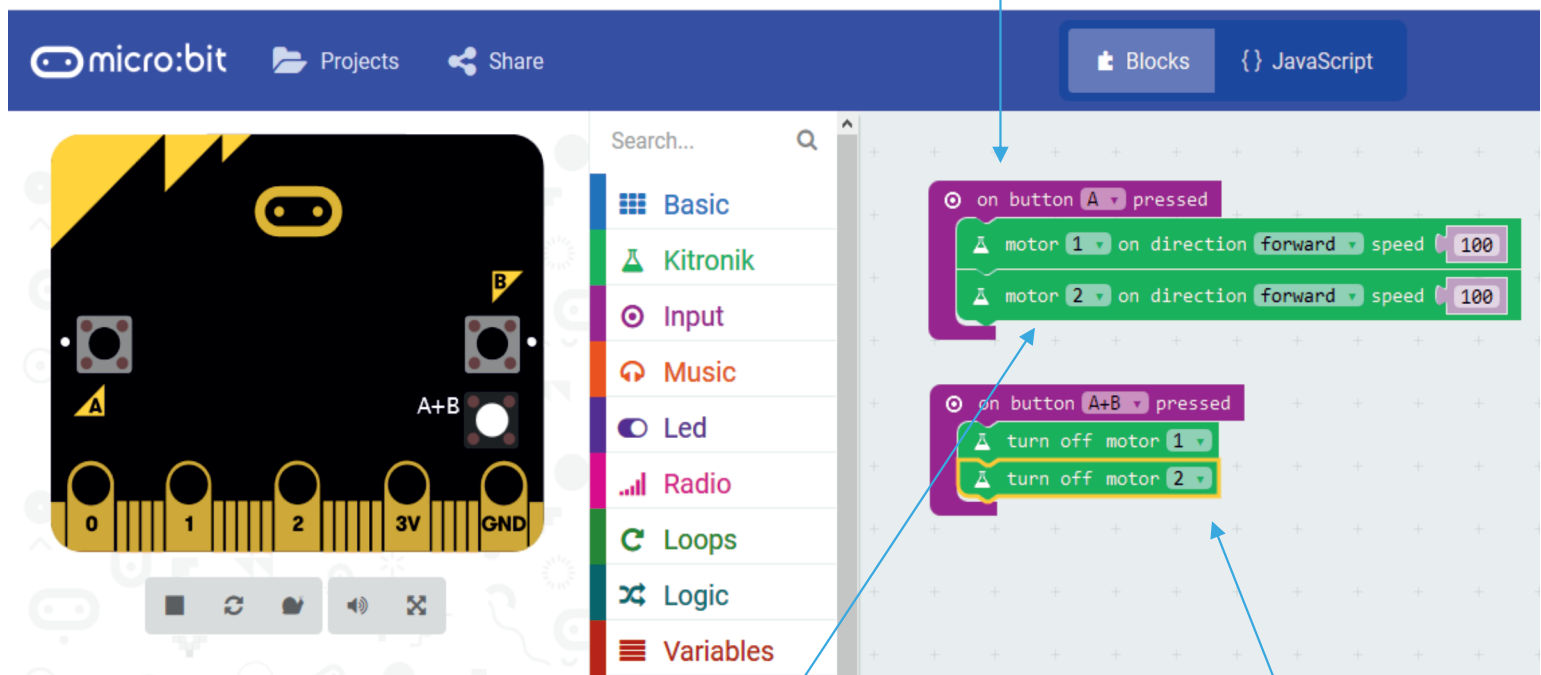
Nyní přistoupíme k **DRUHÉMU** úkolu. K tomu budeme potřebovat dva micro:bity. Jeden bude zapojený v Motor driveru (na vozítku) a druhého s prvním spárujete přes Bluetooth. A jím budete vozítko ovládat.



Zde si můžete vybrat možnost spojení micro:bitů přes Bluetooth.

1. Na stisk tlačítka A rozjedeme vozítko dopředu. Nastavení je pro každý motor zvlášť.
2. Na společné stisknutí tlačítka A+B se vozítko zastaví. Oba motory.

balíček **INPUT** / na tlačítko **A** zmáčknutí
na tlačítko **A+B** zmáčknutí



balíček **KITRONIK - MotorDriver** / motor __ směrem _____ rychlostí _____
motor __ směrem _____ rychlostí _____

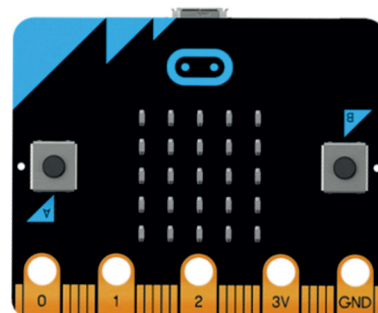
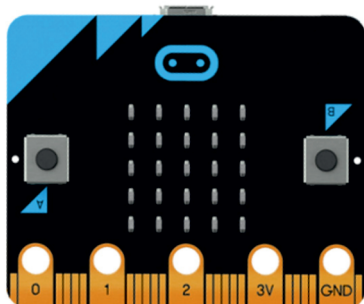
balíček **KITRONIK - MotorDriver** / vypni motor __
vypni motor __

Po ověření správného fungování motorů můžeme přejít k naprogramování micro:bita řízeného micro:bitem za pomoci gyroskopu.

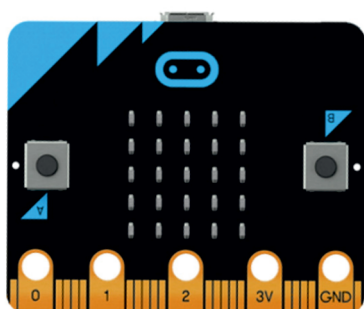
Jak funguje digitální gyroskop

Gyroskopický senzor bývá často zabudován v chytrých mobilních telefonech. Je připojen k akcelerometru. Slouží totiž podobně jako akcelerometr k tomu, aby určoval naklonění a natočení telefonu. Akcelerometr však na rozdíl od gyroskopu měří zrychlení, zatímco gyroskop úhlovou rychlost. Proto je vhodné využívat jejich kombinaci, aby bylo možné přesně určit skutečný pohyb zařízení v prostoru. Systém pak funguje na základě náklonu ovládací destičky.

Náklon destičky pro pohyb robota směrem dopředu



Náklon destičky pro pohyb robota směrem dozadu



Náklon destičky pro pohyb robota ve směru do prava

Náklon destičky pro pohyb zařízení ve směru do leva

