FAKULTA
ELEKTROTECHNICKÁ
ČVUT V PRAZE

BRIAN
PŘÍRUČKA

BriVis

JAN ŘEHÁK, ONDŘEJ VAŠÁTKO

VERZE 0.9

OBSAH

ÚVOD

Stavba robota 3
Příprava prostředí 4
Připojení přes wifi 5
1. Hello World 6
2. Píp píp 7

MOTORY

1. Tak to rozjedem 8
2. Na plný plyn 9
3. Dopředu a dozadu 10
4. Zatáčka 11
5. Točím se točím 12
6. Spirála 14
7. Pilot 15

SENZORY

1. Detekce stisku 17
2. Detekce stisku 18
3. Zpomalit nebo zrychlit 22
4. Zastavit, stát! 24
5. Nenabourej 25
6. Nenabourej 2 27
7. Zastavit, stát! 2 28
8. Gyro senzor 30

POKROČILÉ ÚLOHY

1. Sledování černé čáry 32
2. Sledování černé čáry 2 33

NEŘEŠENÉ ÚLOHY

1. Sledování černé čáry 3 35
2. Překážka na čáře 36
3. Bludiště 37
4. Sumo 38

STAVBA ROBOTA

ÚKOL

Naskenujte QR kód a podle návodu postavte robota.

💡 **TIP**

Pokud vám nefunguje odkaz, můžete jít na adresu https://tinyurl.com/brian-line-follower

PŘÍPRAVA PROSTŘEDÍ

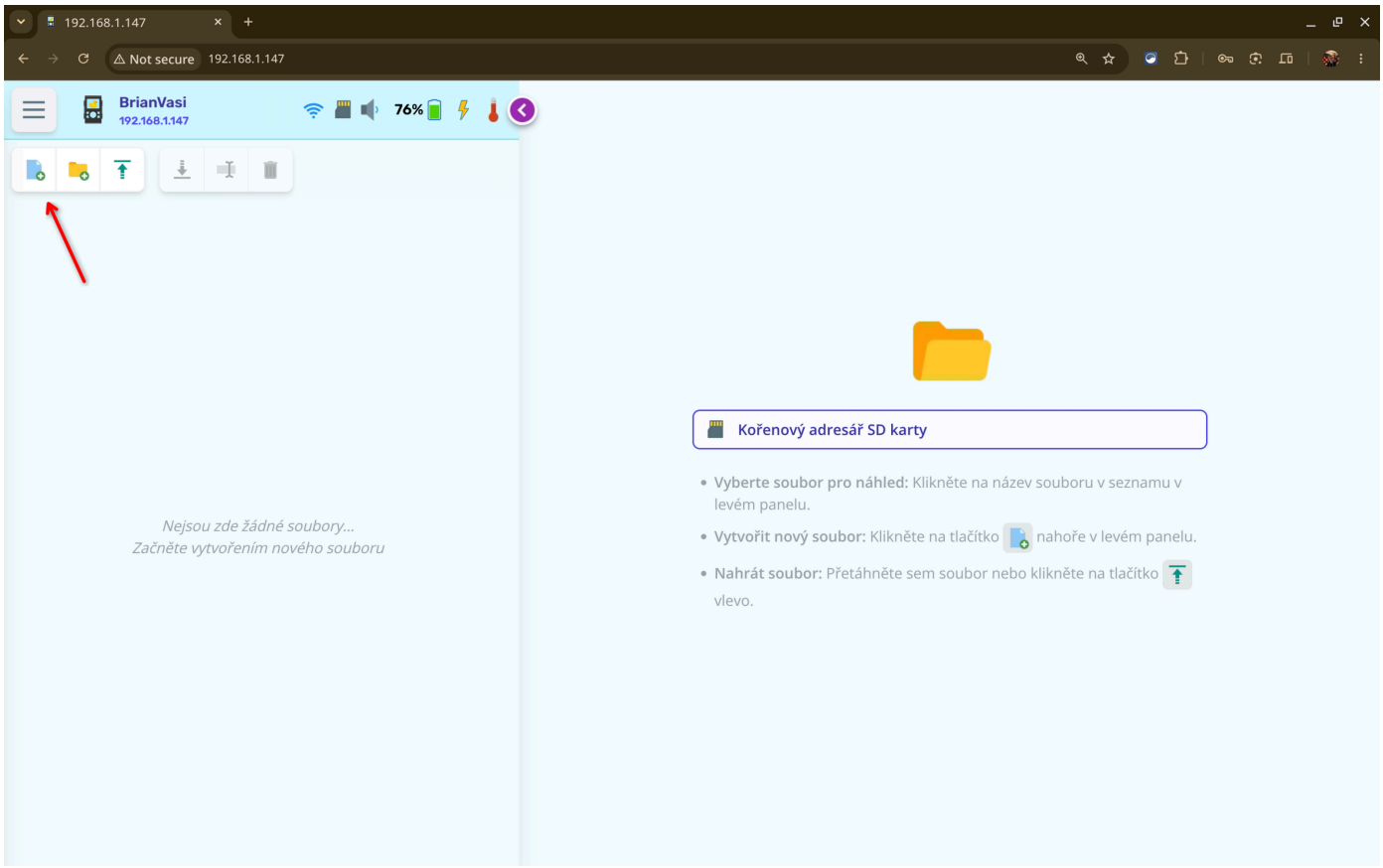
ÚKOL

Připravte si programovací prostředí.

> **TIP**
>
> Pro úplné začátky s programováním doporučujeme začít s BriVis

POSTUP ŘEŠENÍ

1. Připojte Briana ke stejné Wi-Fi síti jako počítač (viz kapitola Připojení přes wifi). Můžete se také počítačem připojit k vlastní síti Briana.
2. Na Brianovi v menu `Settings` → `WiFi config` najděte jeho **IP adresu**.
3. Na počítači otevřete webový prohlížeč (oficiálně je podporován Google Chrome) a do adresního řádku zadejte IP adresu Briana. Otevře se BriVis.
4. Tlačítkem vytvořte nový soubor.



Po vytvoření souboru můžete zvesela programovat.

PŘIPOJENÍ PŘES WIFI

PŘIPOJENÍ K MÍSTNÍ SÍTI

1. V menu `Settings` → `WiFi config` vyberte `Connect to...`, zobrazí se dostupné sítě.
   - Brian musí být v `scan mode`, aby se seznam ukázal.
   - Pokud žádné sítě nevidíte, spusťte skenování přes volby `Start WiFi`, `Disconnect` nebo `Disconnect AP`.
2. Vyberte síť a klikněte na `Connect`.
   - Ikona zámku znamená, že je vyžadováno heslo.
   - Pokud je potřeba, Brian vás vyzve k zadání hesla.
3. **Known (saved) networks** se zobrazují dole na obrazovce.
   - V menu `Manage` (Right button) můžete síť zapomenout (`Forget`) nebo vypnout automatické připojení (`Disable Auto-join`).
   - Brian si pamatuje až 12 sítí.

ZAPNUTÍ VLASTNÍ SÍTĚ

1. V menu `WiFi config` zvolte `Configure AP`.
2. Zadejte `Name (SSID)` a `password`, nebo nastavte otevřenou síť.
3. Poté spusťte (nebo restartujte) `AP`.
   - Brian se odpojí od všech jiných WiFi sítí, pokud byl připojen.
   - V AP režimu se mohou připojit až 3 zařízení.

(RE)BOOT BEHAVIOR

V menu `WiFi config` vyberte `Boot behavior`, kde určíte automatické zapínání WiFi.

BRIVIS PŘES WI-FI

Na připojeném počítači nebo telefonu otevřete webový prohlížeč.
Oficiálně je podporován **Google Chrome**, ale fungují i jiné hlavní prohlížeče.

1. Zadejte **IP address** zobrazenou v `WiFi config`.
2. Otevře se **BriVis**, který nabízí tyto funkce:

Funkce

- File transfer mezi prohlížečem a SD kartou Briana
- Editable file preview: náhled a úprava souborů (některé typy lze editovat jako text), BriVis soubory se otevřou přímo v blokovém editoru
- Program start/control: spouštění libovolných spustitelných souborů na Brianovi přímo z BriVisu, možnost přerušení nebo ukončení
- Firmware update check: pokud má zařízení internet, BriVis upozorní na novou verzi a nabídne odkaz na aktualizační nástroj (samotná aktualizace probíhá přes USB)
- Program console: připojení k běžícímu programu, zobrazení výstupu a posílání příkazů
- Settings: úprava uložených parametrů (jas, WiFi boot behavior atd.)
- Motor demo: vzdálené testování a spouštění motorů z prohlížeče
- Sensor demo: vzdálené čtení hodnot senzorů v prohlížeči

1. HELLO WORLD

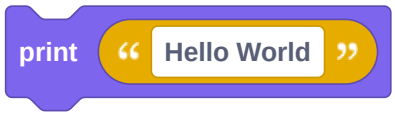
ÚKOL

Úkolem je vypsat text “Hello World” na kostce brian.

> ⚠️ **POZOR**
>
> Nezapomeňte kostku Brian pořádně nabít a zapnout, aby byla připravená! ⚡

> ⚠️ **POZOR**
>
> Zkuste si vytisknout i jiné fráze, ale pozor, Brian neumí tisknout české znaky, musíte proto psát bez háčků a čárek

POSTUP ŘEŠENÍ



1. V menu BriVisu klikněte na zelenou šipku pro spuštění
2. Koukněte se na obrazovku kostky

A je to! Kostka Brian by teď měla říct „Hello World“. Skvělá práce, programátore! 🥳

2. PÍP PÍP

ÚKOL

Úkolem je napsat program, který zahraje tón.

> 💡 **TIP**
>
> Pro každou úlohu si doporučujeme vytvořit nový soubor

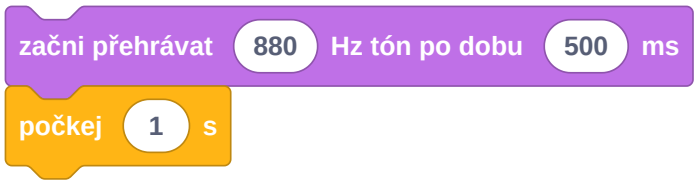
> 💡 **NÁPOVĚDA**
>
> Použijte blok `začni přehrávat __ Hz tón po dobu __ ms`. Do mezer doplňte dvě čísla: první určuje výšku tónu a druhé délku tónu v milisekundách (1000ms = 1s).

> 💡 **NÁPOVĚDA**
>
> Po tom, co necháme hrát tón nesmíme předčasně ukončit program než tón dohraje. Použijeme tedy blok `počkej __ s`, který chvíli počká, než tón dohraje.

> 💡 **TIP**
>
> Doporučujeme příjemnou výšku tónu 880, můžete ale experimentovat s dalšími tóny.

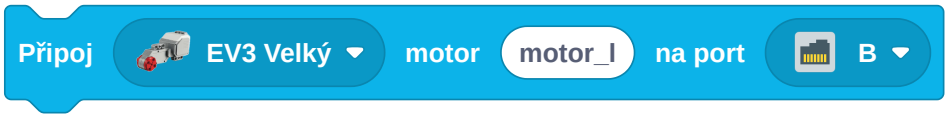
> ⚠️ **POZOR**
>
> Zkontrolujte, že máte na kostce zapnutou hlasitost. V menu jděte do `Settings` -> `Sound volume` a nastavte hlasitost.

POSTUP ŘEŠENÍ



PŘIPOJENÍ MOTORU

Abychom mohli s motorem pracovat, musíme si ho pojmenovat.



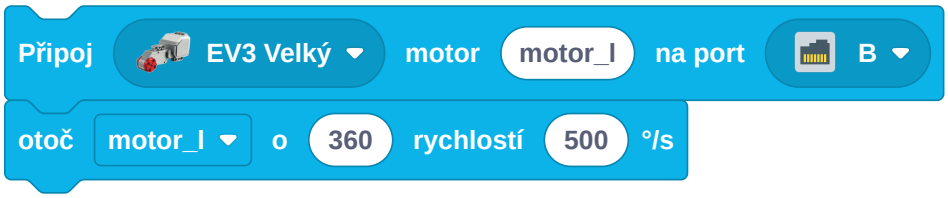
ÚKOL

Naprogramovat jeden motor tak, aby se otočil o 360° rychlostí 500°/s.

NÁPOVĚDA

Použijte blok `otoč __ o __ rychlostí __ °/s`

POSTUP ŘEŠENÍ



2. NA PLNÝ PLYN

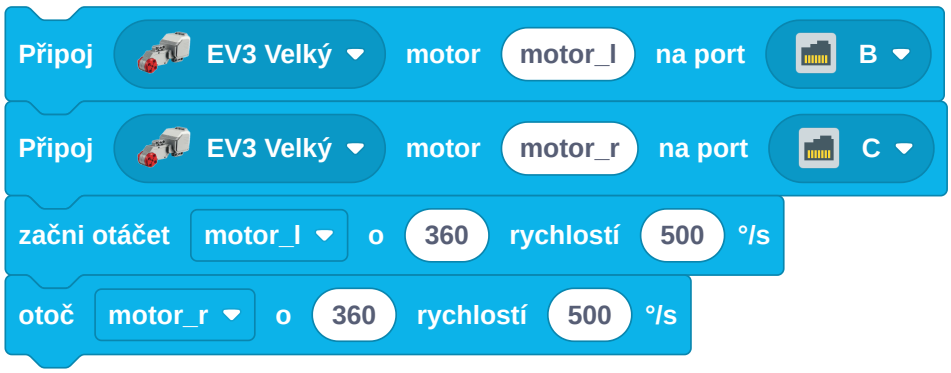
ÚKOL

Naprogramovat robota tak, aby popojel vpřed o jedno otočení koly.

NÁPOVĚDA

Pro rozjetí obou motorů najednou, je potřeba pro první motor použít blok `začni otáčet __ o __ rychlostí __ °/s`.
Pro druhý motor použít stejný blok jako v předchozím úkolu.

POSTUP ŘEŠENÍ



TIP

Maximální rychlost pro velký motor je `1050 deg/s` (stupňů za sekundu) a `1560 deg/s` pro střední motor

3. DOPŘEDU A DOZADU

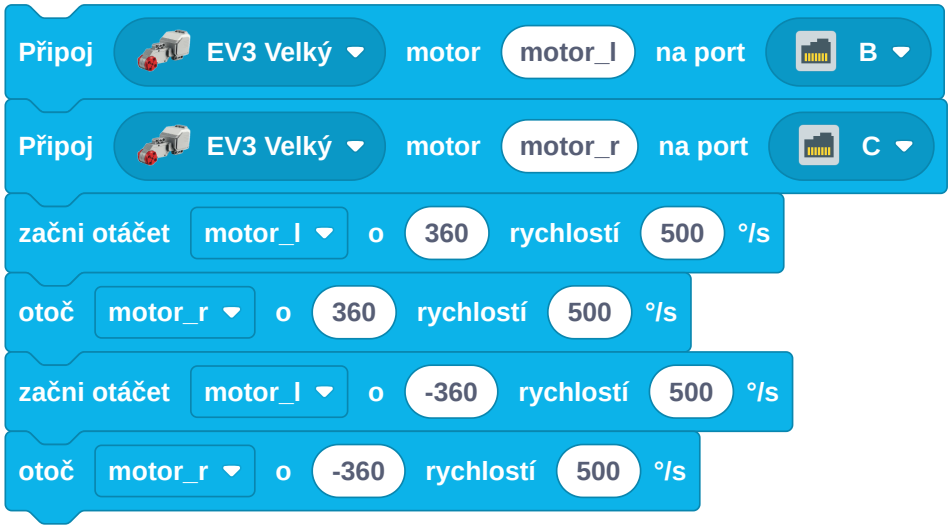
ÚKOL

Naprogramovat robota tak, aby popojel dopředu a poté dozadu.

NÁPOVĚDA

Záporné číslo v políčku rychlosti popožene robota dozadu.

POSTUP ŘEŠENÍ



4. ZATÁČKA

ÚKOL

Naprogramovat robota tak, aby se otočil do pravého úhlu.

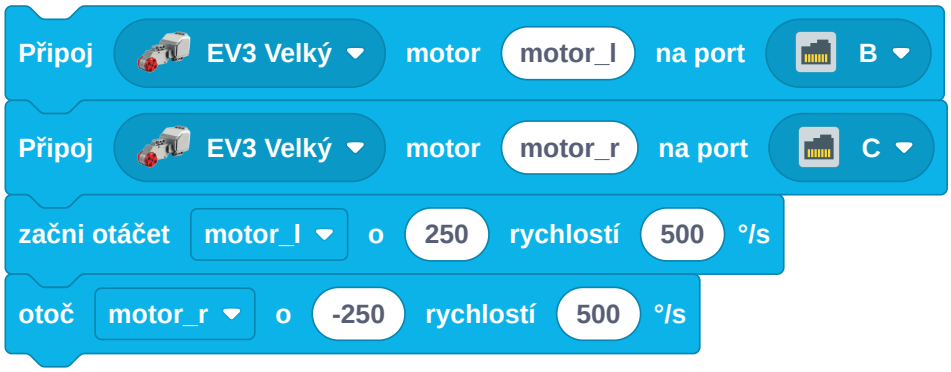
NÁPOVĚDA

Aby se robot otočil, musíme jedním motorem otočit na jednu stranu a druhým na druhou.

VYZKOUŠEJTE SI

Experimentujte s rychlostí jednotlivých motorů, aby robot vytvořil různě velké zatáčky.

POSTUP ŘEŠENÍ



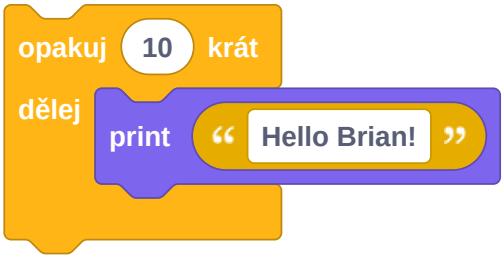
TIP

Hodnota `250` je jen odhad. Vyzkoušejte, o kolik stupňů se musí kola otočit, aby se robot otočil přesně o 90°.

5. TOČÍM SE TOČÍM

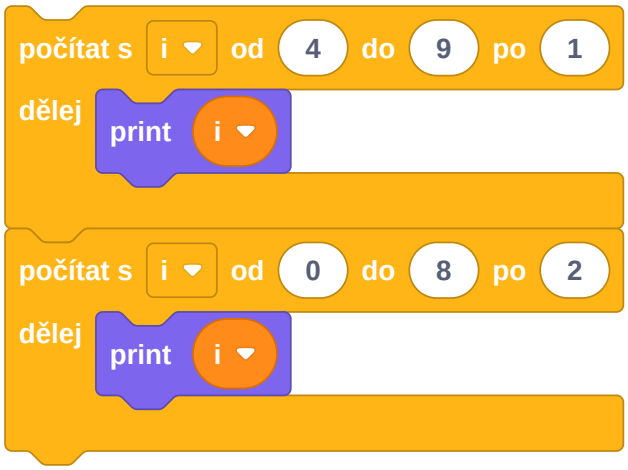
PROČ OPAKOVAT PŘÍKAZY

Představte si, že chcete robotovi 10× říct „Hello Brian!“. Místo skládání stejného bloku pořád dokola mu můžeme dát úkol zopakovat ho za nás. Slouží k tomu blok `opakuj __ krát`. Všechny bloky uvnitř něj (u slova `dělej`) se zopakují tolikrát, kolik je napsáno v políčku.



ZKOUŠÍME ČÍSLA

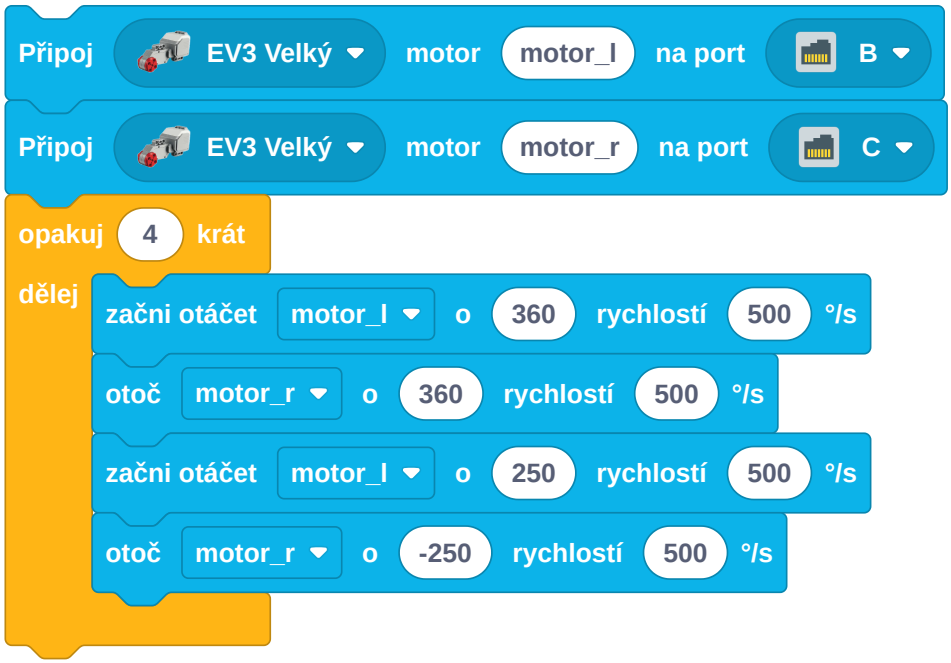
Když potřebujeme vědět, po kolikáté se cyklus opakuje, použijeme blok `počítat s i od __ do __ po __`. Proměnná `i` postupně nabývá čísel od začátku až po konec rozsahu, v BriVisu včetně posledního čísla. Můžeme si hrát s tím, kde začneme, kde skončíme nebo o kolik budeme přeskakovat.



ÚKOL

Úkolem je naprogramovat robota tak, aby objel čtverec. Použijte *for cyklus*.

POSTUP ŘEŠENÍ



TIP

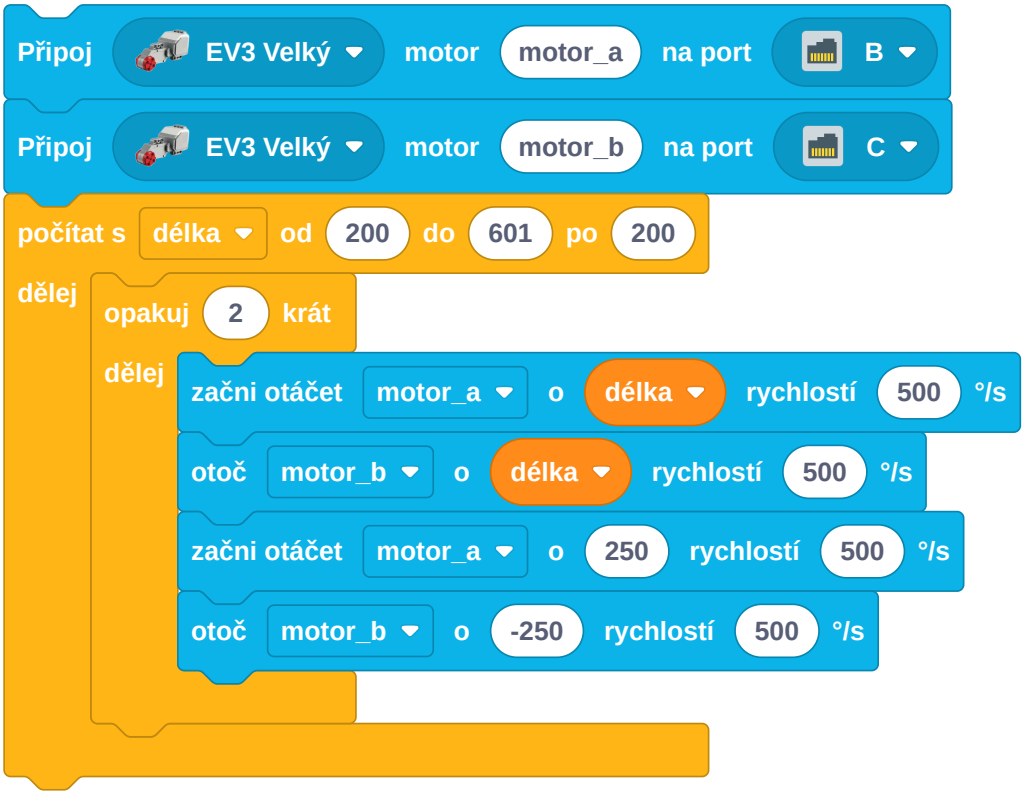
Hodnota `250` je jen odhad. Vyzkoušejte, o kolik stupňů se musí kola otočit, aby se robot otočil přesně o 90°.

6. SPIRÁLA

ÚKOL

Naprogramovat robota tak, aby se točil do spirály. Použijte vnořený *for cyklus*.

POSTUP ŘEŠENÍ



TIP

Hodnota `250` je jen odhad. Vyzkoušejte, o kolik stupňů se musí kola otočit, aby se robot otočil přesně o 90°.

PILOT

Doteď jsme každý motor řídili zvlášť a počítali, o kolik stupňů se má otočit. Jednodušší je použít **pilota**. Ten řídí oba motory najednou: řekneme mu, kolik milimetrů má ujet nebo o kolik stupňů se má otočit robot, a pilot si sám spočítá, jak mají jet motory.

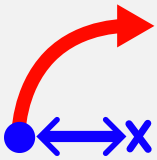
Aby to spočítal, musí znát dva rozměry robota: **vzdálenost mezi koly** (měří se od středu jednoho kola ke středu druhého, u našeho robota 110 mm) a **průměr kol** (40 mm).

Pilota vytvoříme blokem `Vytvoř pilota`, kde vybereme levý a pravý motor a vyplníme vzdálenost mezi koly a průměr kol. Pak už robota řídíme bloky `jeď s pilotem` a `otáčej s pilotem`.

OTÁČENÍ, OBLOUK A ZATÁČENÍ

Pilot umí zatáčet třemi způsoby. Liší se tím, co mu zadáme: o kolik se má robot otočit, po jak velké kružnici má jet, nebo jak moc má zatáčet.

Otáčení otočí robota o zadaný úhel. S poloměrem `0` se točí na místě kolem svého středu.



Oblouk jede po kružnici se zadaným poloměrem, dokud neujede zadanou vzdálenost. Záporný poloměr zatáčí doleva.

Zatáčení funguje jako volant: `0` jede rovně, `0.5` jedno kolo stojí, `1` otáčí na místě. Záporné číslo zatáčí doleva.

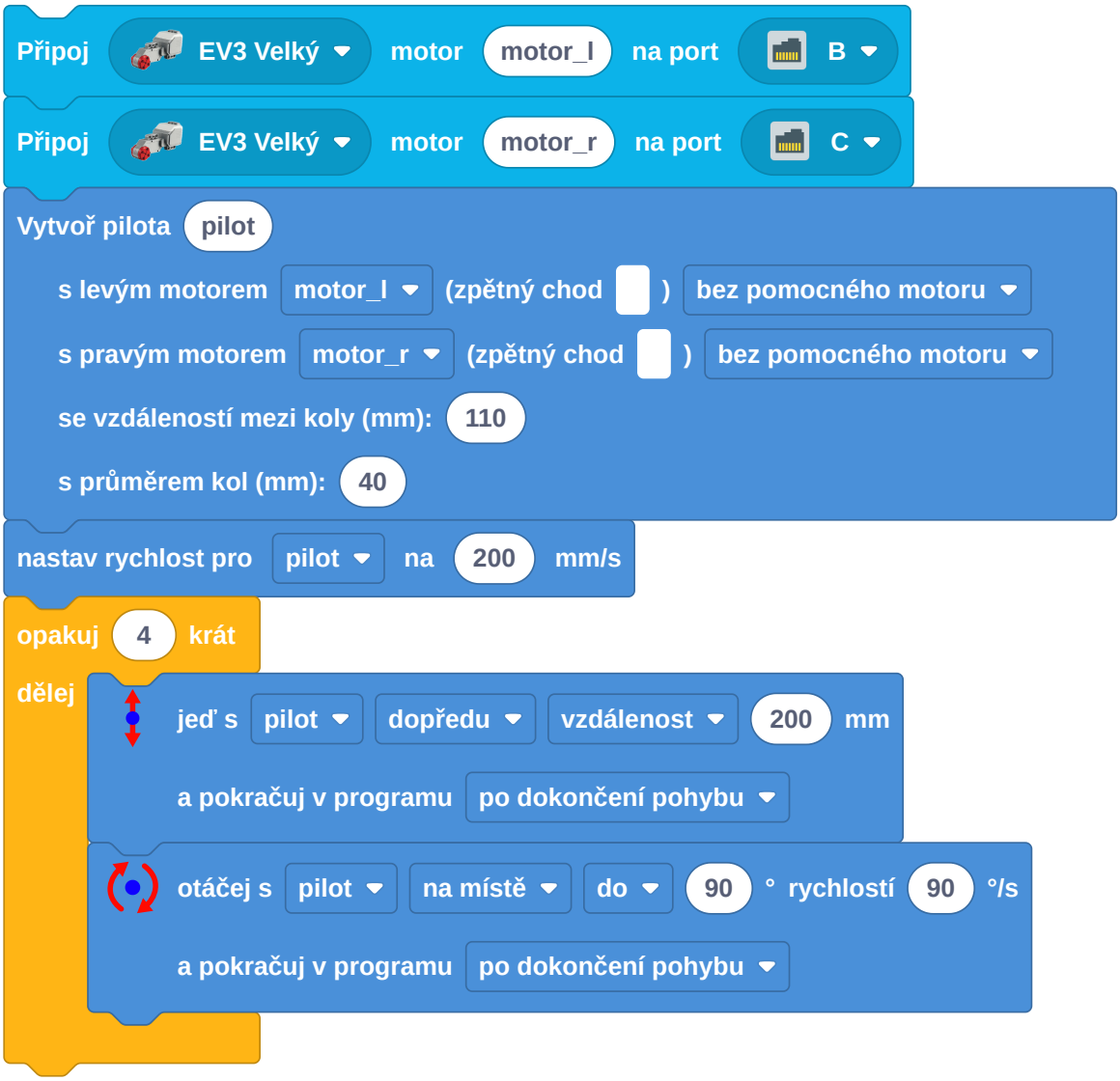
V BriVisu: bloky `otáčej s pilotem`, `jeď s pilotem po kružnici` a `jeď s pilotem se zatáčením`.

> 💡 **TIP**
>
> Od teď budeme jezdit s pilotem. Motory zvlášť použijeme jen tam, kde potřebujeme každé kolo řídit jinak, například u regulátorů.

ÚKOL

Objeďte znovu čtverec z úlohy 5, tentokrát s pilotem. Strana čtverce bude 20 cm.

POSTUP ŘEŠENÍ



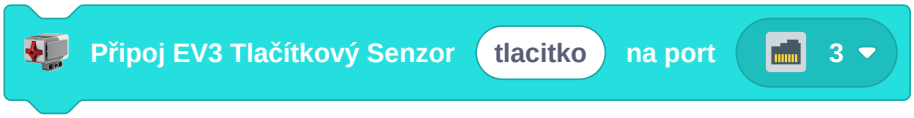
1. DETEKCE STISKU

TLAČÍTKO

První senzor, který si ukážeme, je `TouchSensor`.

Robot dokáže poznat, jestli je zmáčknuté. Díky tomu s ním můžete třeba spouštět různé funkce nebo zjistit, že robot narazil do zdi.

PŘIPOJENÍ TLAČÍTKA



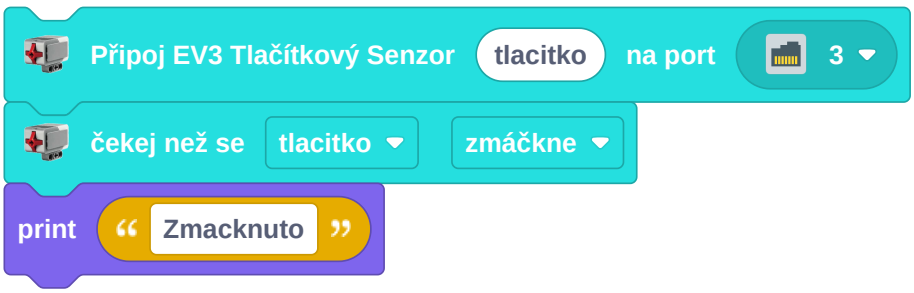
ÚKOL

Počkejte na zmáčknutí tlačítka a potom napište na obrazovku Briana `Zmacknuto`.

NÁPOVĚDA

Použijte blok `čekej než se __ zmáčkne`

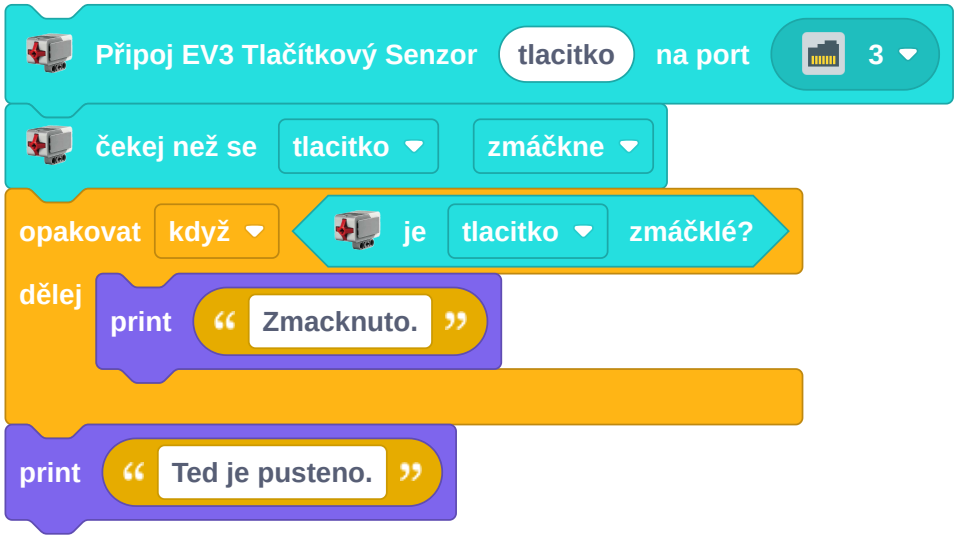
POSTUP ŘEŠENÍ



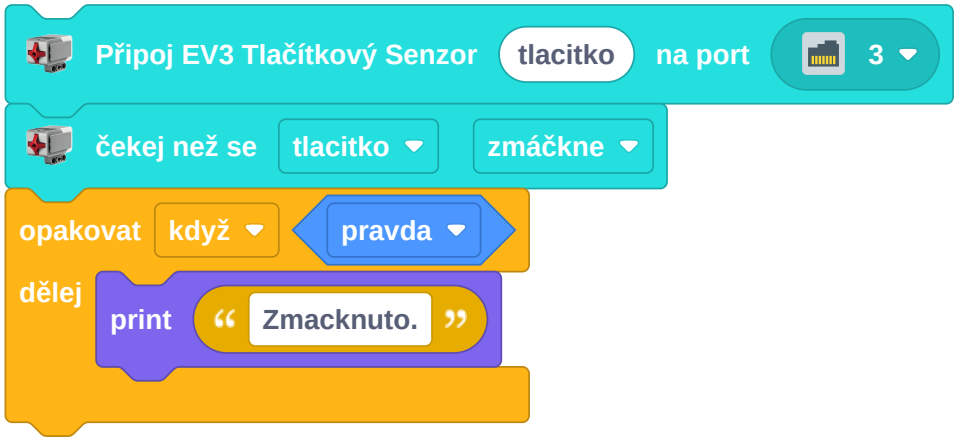
WHILE CYKLUS

V některých případech dopředu nevíme, kolikrát budeme muset nějaký příkaz nebo nějaké příkazy vykonat, a nemůžeme proto použít *for-cyklus*, který má pevně daný počet opakování. Pro tyto případy můžeme použít *while* cyklus, který opakuje příkazy ve svém těle dokud je splněna nějaká podmínka.

V BriVisu k tomu slouží blok `opakovat když`. Bloky uvnitř (u slova `dělej`) se opakují tak dlouho, dokud platí podmínka. Ukážeme si to na jednoduchém příkladu, kde se na obrazovku Briana bude tisknout `Zmacknuto.` dokud budeme držet tlačítko.



Může se stát, že chceme aby cyklus běžel pořád, bez ukončení. Do podmínky pak vložíme blok `pravda`.



VYZKOUŠEJTE SI

V následujících úlohách se může stát, že *while* cyklus nepůjde opustit a program se sám nezastaví. Zastavit ho ale můžete stisknutím obou horních tlačítek kostky současně.

VYZKOUŠEJTE SI

Všechny programy zapsané pomocí *for cyklu* je možné přepsat i pomocí *while* cyklu, ne obráceně.

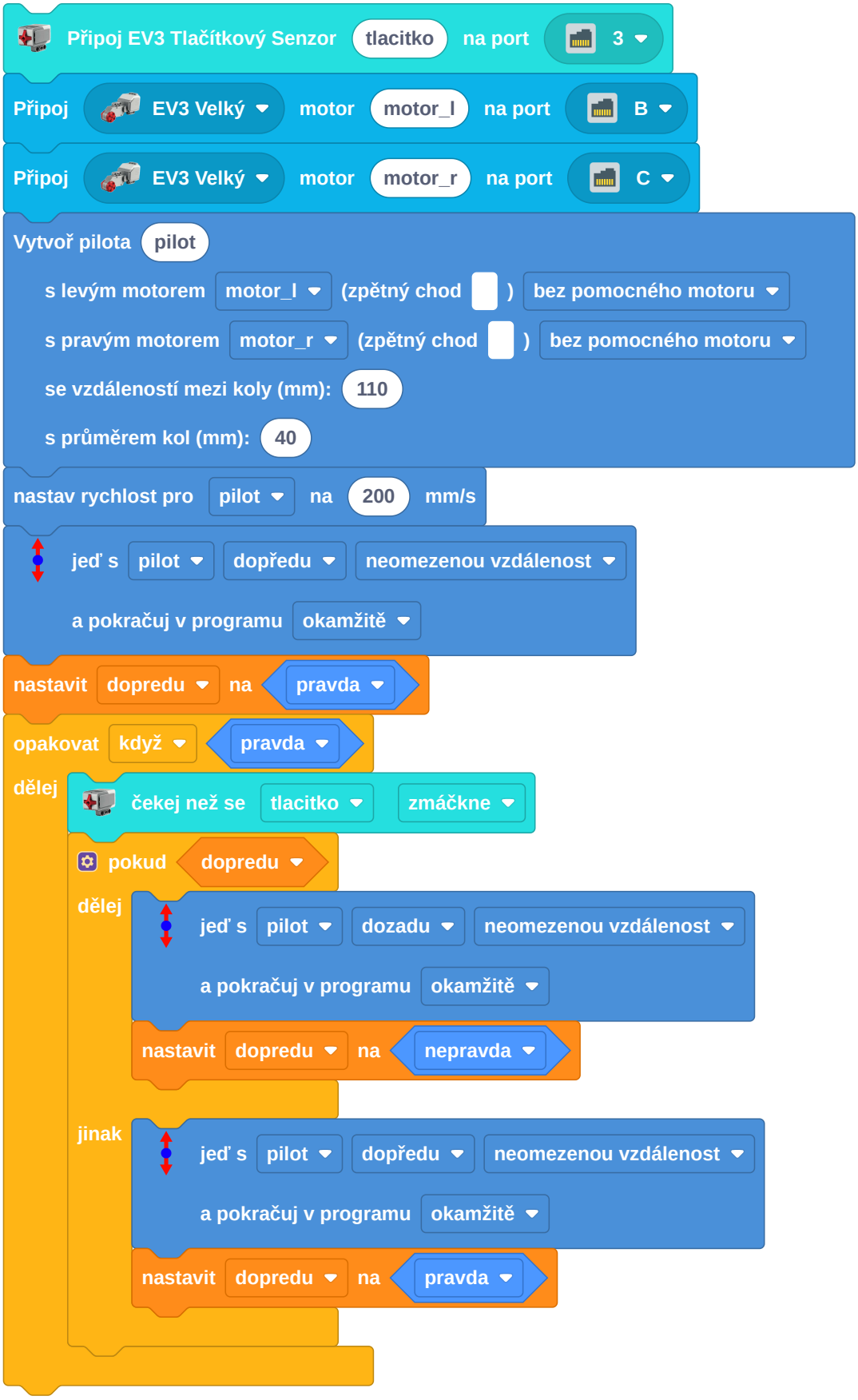
ÚKOL

Rozjeďte se dopředu a po zmáčknutí tlačítka otočte směr jízdy.

💡 NÁPOVĚDA

Bude potřeba proměnná, která si bude pamatovat, jakým směrem robot jede.

POSTUP ŘEŠENÍ



💡 **TIP**

Použijte blok `čekej než se ___ zmáčkne`

3. ZPOMALIT NEBO ZRYCHLIT

ÚKOL

Rozjeďte se dopředu. Zmáčknutím jednoho tlačítka by se měl robot zrychlit a zmáčknutím druhého zase zpomalit.

⚠️ **POZOR**

Rychlost pilota zadáváme v milimetrech za sekundu. S našimi koly robot zvládne nejvýše asi `360 mm/s` a rychlost by neměla klesnout až na `0`.

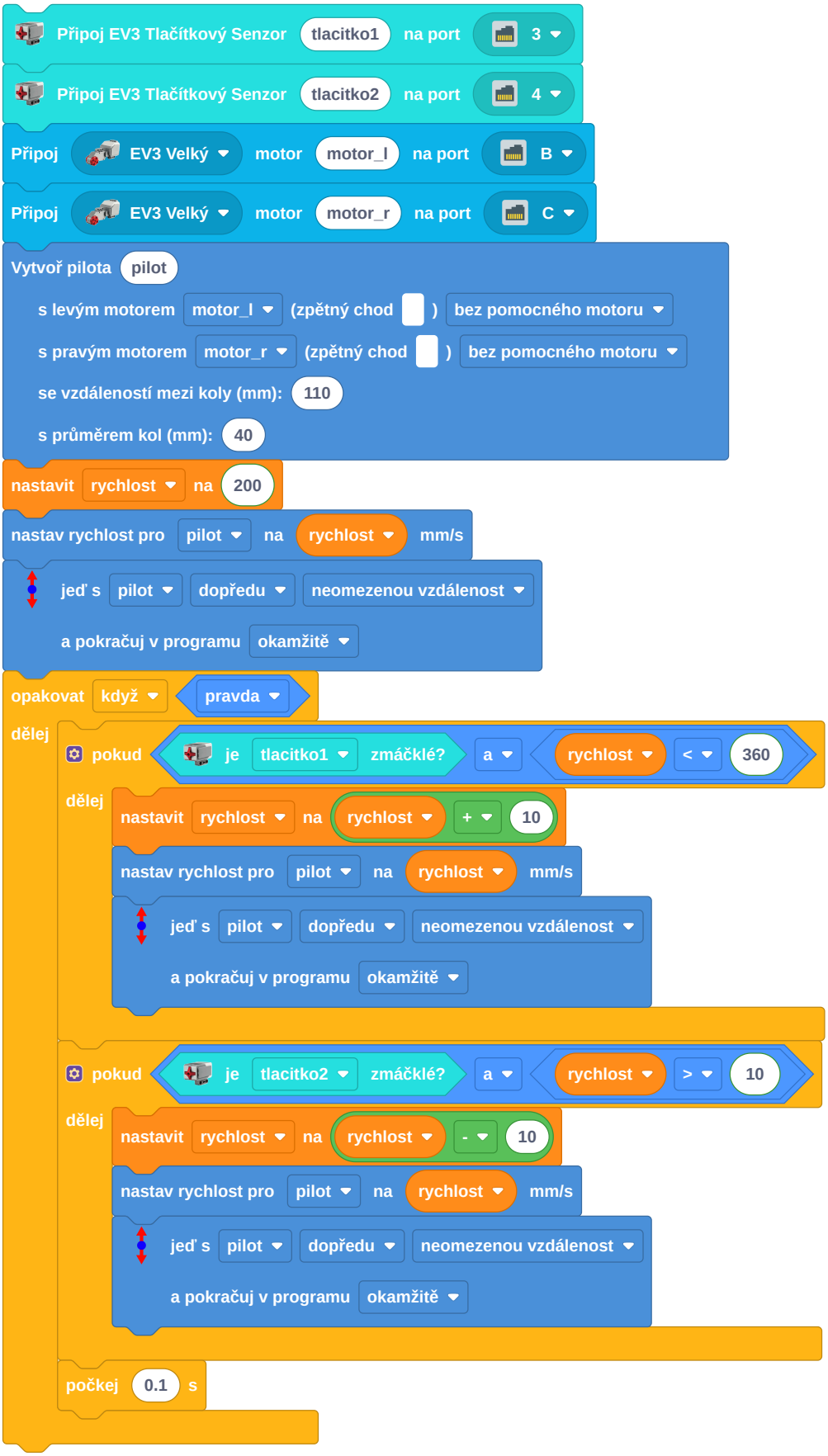
⚠️ **POZOR**

Ve smyčce použijte blok `počkej 0.1 s`, aby se jedno stisknutí tlačítka nezaznamenalo jako několik naráz.

💡 **NÁPOVĚDA**

Budete potřebovat proměnnou, která si bude pamatovat rychlost jízdy.

POSTUP ŘEŠENÍ



4. ZASTAVIT, STÁT!

ÚKOL

Upevněte tlačítko na předek robota. Rozjeďte se dopředu a jakmile tlačítko narazí na zeď, měl by robot automaticky zastavit.

POSTUP ŘEŠENÍ

ULTRASONICKÝ SENZOR

Dost často potřebujeme měřit vzdálenost. K tomuto účelu slouží `UltrasonicSenzor`.

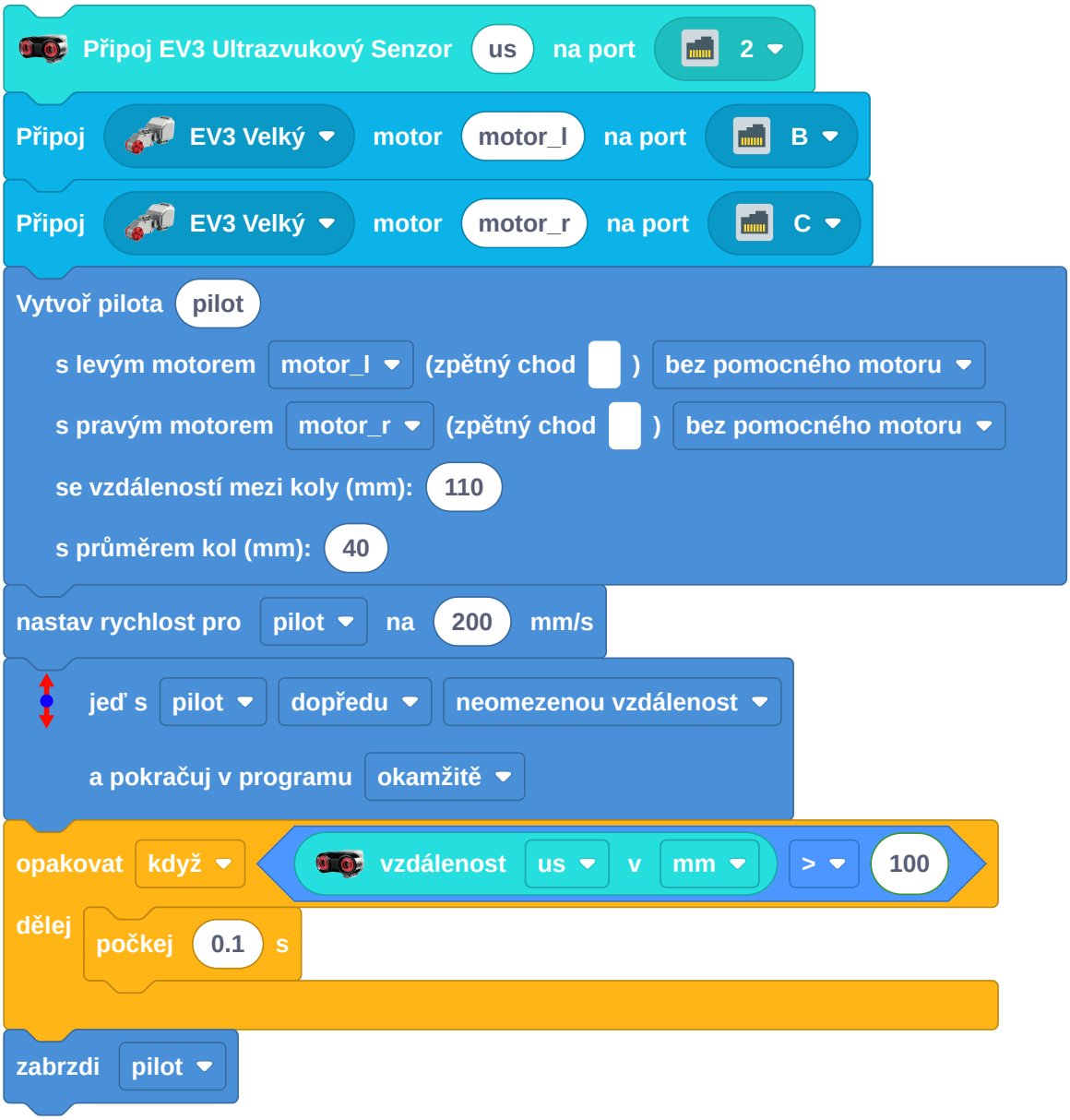
Ten pomocí zvuku a jeho zpětného odražení dokáže měřit vzdálenost.

> ⚠️ **POZOR**
>
> Protože senzor používá zvuk, nemusí dávat ideální výsledky při odrážení zvuku od látky.

ÚKOL

Rozjeďte se dopředu a měřte vzdálenost před robotem. Když je vzdálenost menší než `100 mm` (10 cm), zastavte, aby robot nenaboural.

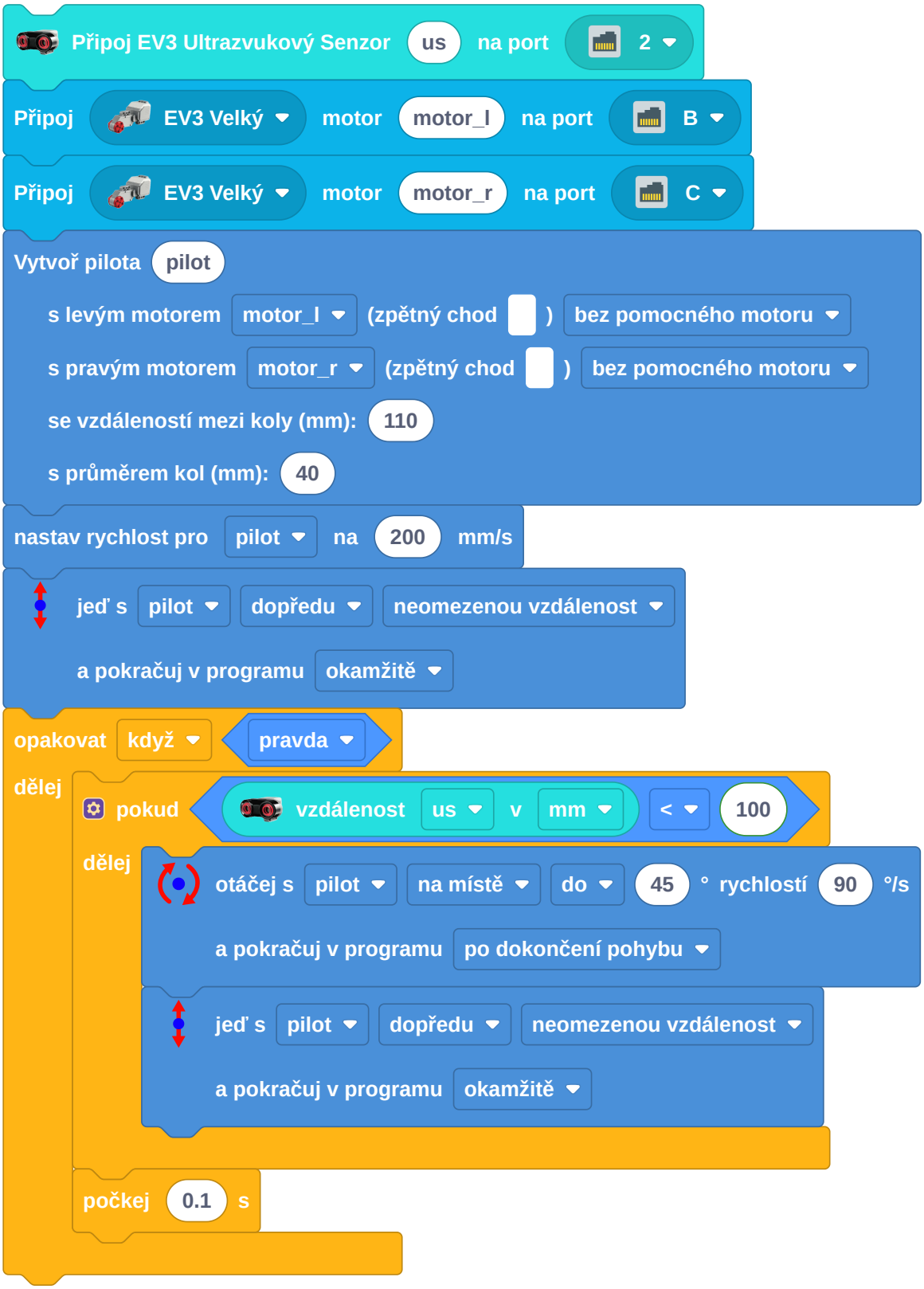
POSTUP ŘEŠENÍ



ÚKOL

Rozjeďte se dopředu a měřte vzdálenost před robotem. Když je vzdálenost menší než `100 mm` (10 cm), pootočte robota a objeďte překážku.

POSTUP ŘEŠENÍ



BAREVNÝ SENZOR

Třetí senzor co si ukážeme je `ColorSensor`.

- Senzor vydává světlo (červené, nebo bílé) směrem dolů či dopředu A poté měří, kolik toho světla se od něčeho odrazí. To je `reflected light` (odražené světlo).
- Může také měřit `ambient light` (okolní světlo), tedy jak je světlý prostor kolem senzoru bez aktivního osvětlení.
- Senzor také umí rozeznat barvy – třeba červenou, zelenou, modrou, černou, bílou a další.

⚠️ **POZOR**

Povrch materiálu hodně ovlivňuje měření – lesklé plochy odráží světlo víc, matné méně.

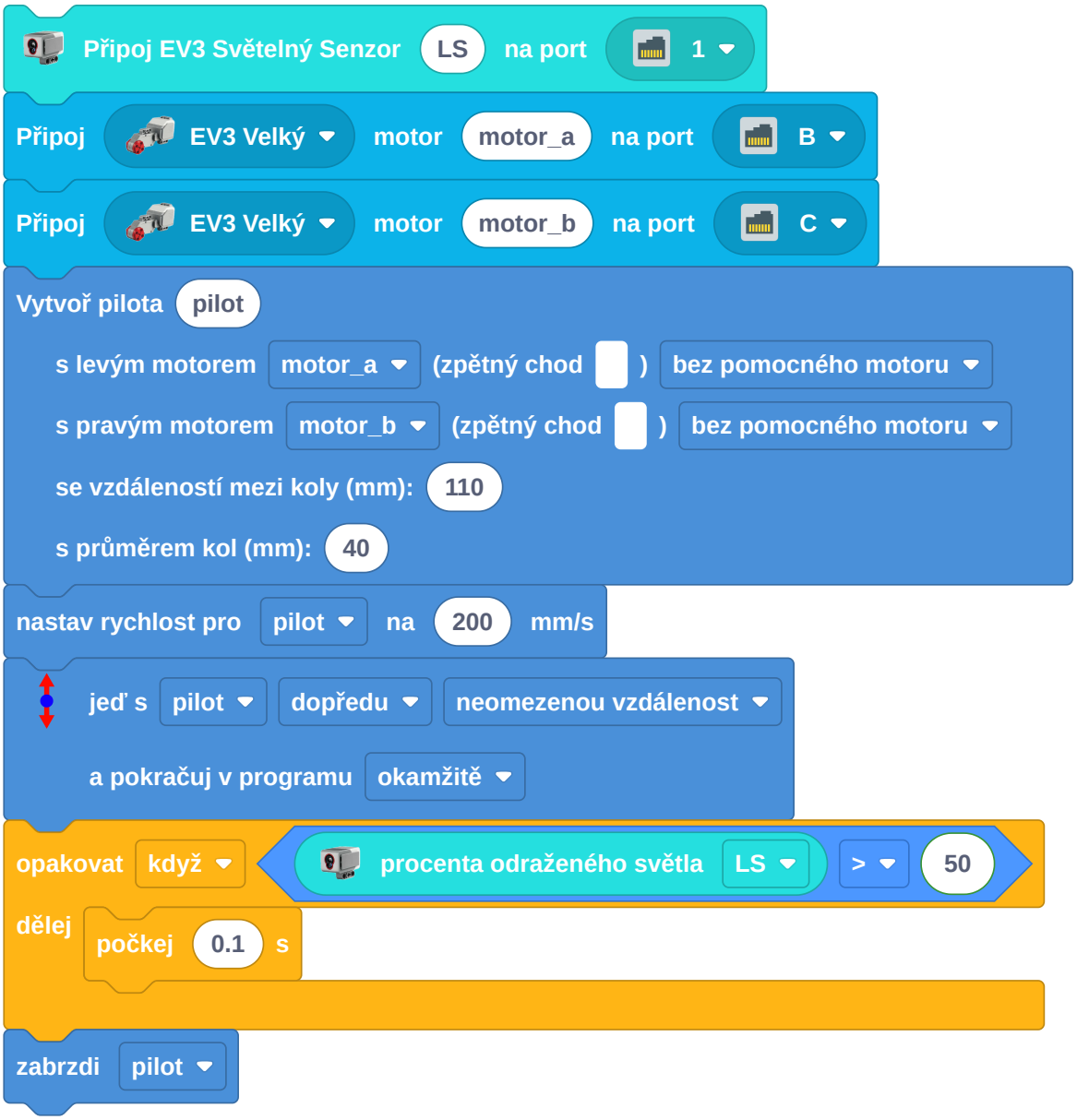
💡 **TIP**

Umístěte světelný senzor co nejblíž k podlaze, aby nepropustilo moc světla bočními cestami.

ÚKOL

Jeďte dopředu, dokud barevný senzor nenajde černou čáru, potom automaticky zastavte.

POSTUP ŘEŠENÍ



8. GYRO SENZOR

GYROSKOPICKÝ SENZOR

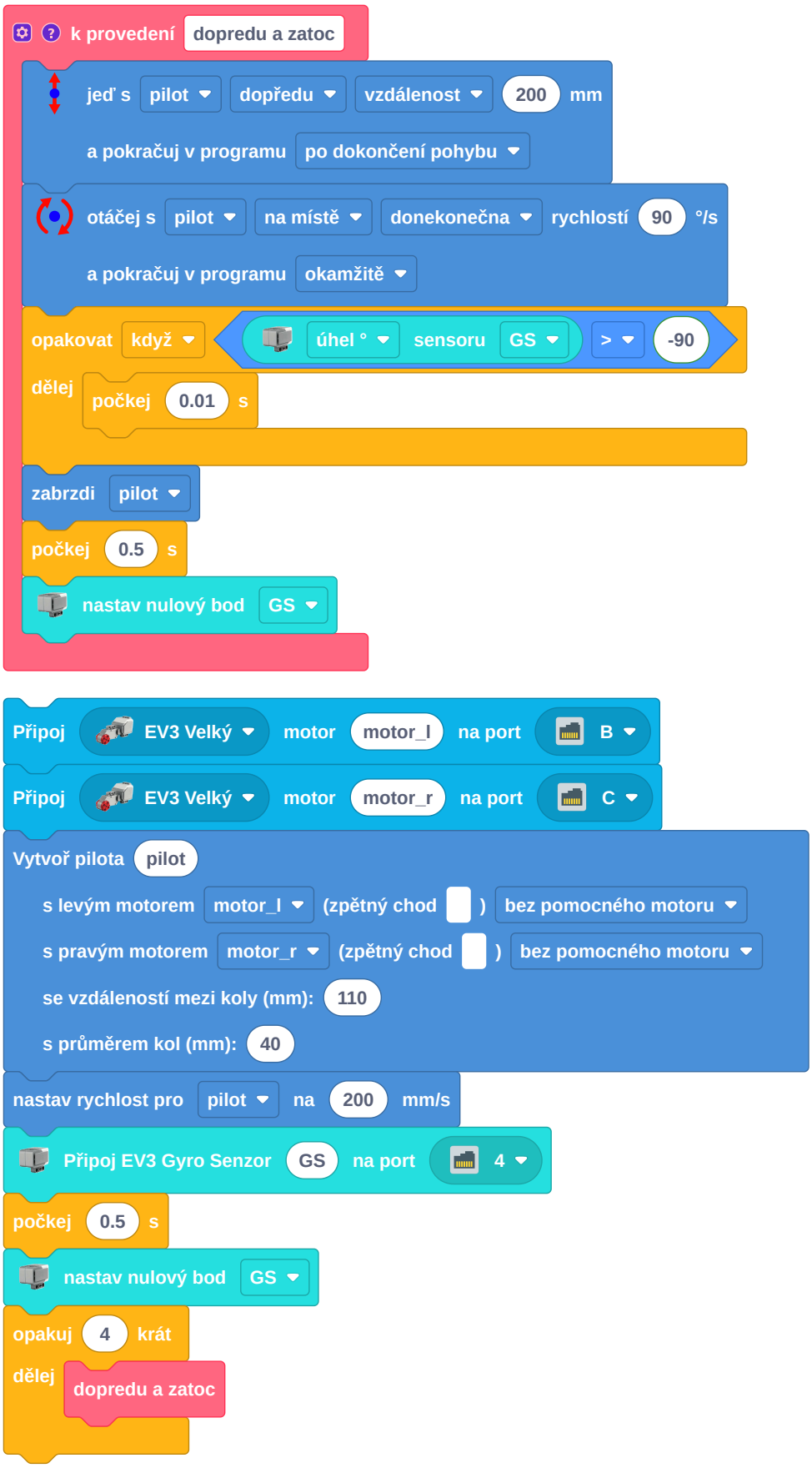
Posledním senzorem který si ukážeme je `GyroSensor`. Ten slouží k měření otáčení robota.

> ⚠️ **VAROVÁNÍ**
>
> Pokud `GyroSensor`, tak je třeba zkalibrovat. V `Demo programs` -> `Sensor view` -> `Port X` -> `Modes` (tlačítko vpravo) -> Na konci seznamu `Reboot (fix drift)`
> Při kalibraci se s Brianem nesmí hýbat, jinak se na na gyru začne nasčítávat úhel.

ÚKOL

Udělat funkci, která jede s robotem rovně, poté se pomocí gyra otočí o 90°. A tuto funkci zopakovat 4x po sobě.

POSTUP ŘEŠENÍ



1. SLEDOVÁNÍ ČERNÉ ČÁRY

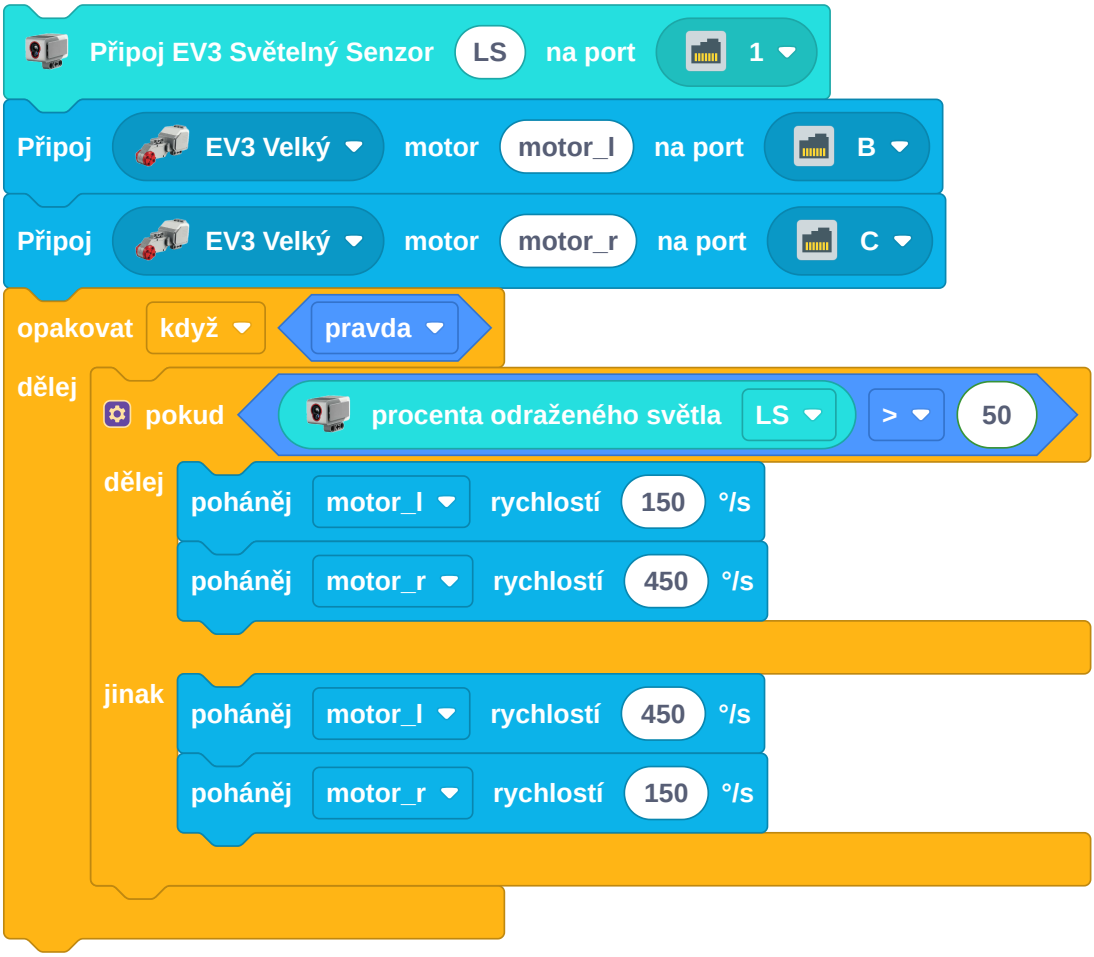
ÚKOL

Naprogramujte robota, co bude sledovat černou čáru.

NÁPOVĚDA

Robot nemusí sledovat střed čáry. Skvěle se mu bude sledovat hranice čáry, kde uvidí černou na jedné straně a bílou na druhé. Program může mít logiku, pokud vidím černou, zatoč doprava, pokud bílou, zatoč doleva. Robot přitom pořád jede dopředu, jen jedno kolo pojede rychleji než druhé, takže opíše oblouk a nebude se točit na místě.

POSTUP ŘEŠENÍ



2. SLEDOVÁNÍ ČERNÉ ČÁRY 2

P REGULACE

Už umíme naprogramovat robota tak, aby následoval černou čáru. Akorát vidíme, že robot jezdí zig-zag. I když by měl jet rovně se tak se kroutí. Proto použijeme **P regulaci** (proporcionální regulaci). Je to chytrý způsob, jak robota naučit, že čím víc se odchýlí od čáry, tím víc se musí stočit zpátky. Pokud je odchylka malá, opraví se jen trochu. Pokud velká, zatočí víc.

Jak na to v programu:

1. **Najděte středovou hodnotu** senzoru – mezi černou a bílou (např. průměr).
2. **Změřte aktuální hodnotu** senzoru.
3. **Spočítejte chybu**: `chyba = aktuální_hodnota - středová_hodnota`.
4. **Vynásobte chybu zesílením (Kp)**: `oprava = Kp * chyba`.
5. **Upravte rychlost motorů**:
   - levý motor = základní_rychlost – oprava
   - pravý motor = základní_rychlost + oprava

Úkolem bude vyzkoušet různé hodnoty `Kp`, dokud robot nepojede plynule.

📝 VYZKOUŠEJTE SI

Pozorujte, co se stane, když je `Kp` příliš malé nebo příliš velké.

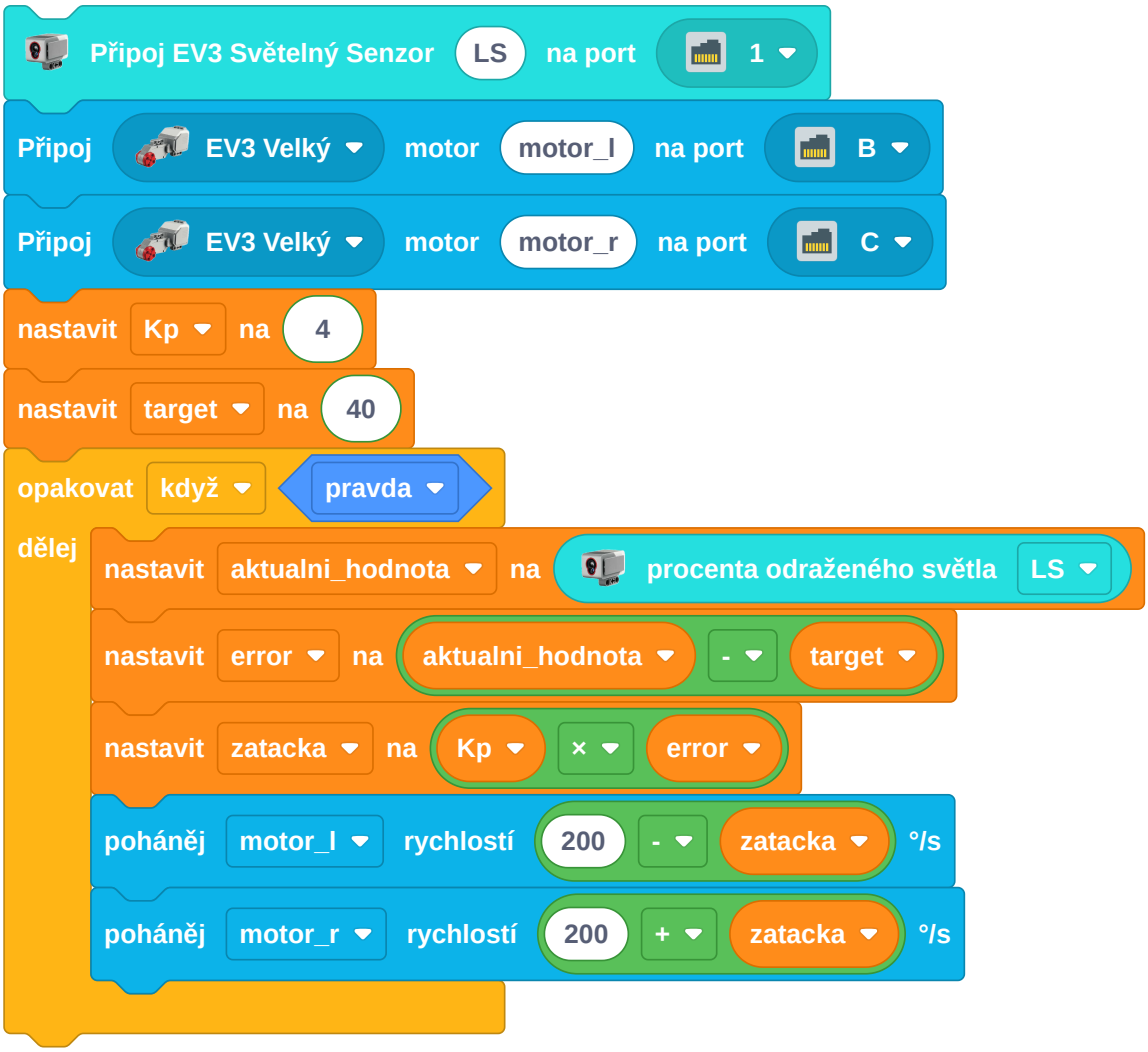
💡 NÁPOVĚDA

Středovou hodnotu senzoru můžete naměřit po zapojení barevného senzoru v sekci `Demo programs` -> `Sensor view`.

ÚKOL

Naprogramujte robota, co bude sledovat černou čáru pomocí P regulátoru.

POSTUP ŘEŠENÍ



PID REGULACE

P regulace už zvládne držet čáru, ale někdy se robot rozkmitá nebo nejede plynule. Proto přidáme **PID regulaci**:

- **P (proporcionální)** – okamžitá oprava podle chyby.
- **I (integrální)** – sčítá chyby a vrací robota zpět, když je dlouhodobě mimo.
- **D (derivační)** – sleduje rychlost změny chyby, aby se nepřekmital.

Jak na to v programu:

1. `chyba = aktuální - středová`
2. `suma_chyb += chyba`
3. `delta_chyby = chyba - předchozí`
4. `P = Kp * chyba`
   `I = Ki * suma_chyb`
   `D = Kd * delta_chyby`
5. `oprava = P + I + D`
6. levý = základ − oprava
   pravý = základ + oprava

> 💡 **TIP**
>
> Začněte jen s **P** (Ki=0, Kd=0).
> `Kp` moc velké → kmity.
> `Ki` moc velké → pomalé „plavání".
> `Kd` moc velké → trhaná jízda.

S PID regulací jede robot plynuleji a chytřeji.

ŘEŠENÍ

> 📝 **ŘEŠENÍ**
>
> Řešení této úlohy najdete pouze v Materiálech na https://gitlab.fel.cvut.cz/rehakj16/brian-prirucka-assets.

2. PŘEKÁŽKA NA ČÁŘE

ÚKOL

Naprogramujte robota, co bude sledovat černou čáru, když na čáře bude překážka, objede ji a vrátí se na černou čáru za překážkou.

ŘEŠENÍ

📝 ŘEŠENÍ

Řešení této úlohy najdete pouze v Materiálech na https://gitlab.fel.cvut.cz/rehakj16/brian-prirucka-assets.

ÚKOL

Naprogramujte robota, který najde cestu z bludiště z kostek.

ŘEŠENÍ

📝 ŘEŠENÍ

Řešení této úlohy najdete pouze v Materiálech na https://gitlab.fel.cvut.cz/rehakj16/brian-prirucka-assets.

4. SUMO

ÚKOL

Uspořádejte sumo zápasy robotů. Stačí vyhrazené pole na zemi. Vždy se postaví dva roboti proti sobě a kde zůstane v poli déle, vyhrál.

ŘEŠENÍ

📝 ŘEŠENÍ

Řešení této úlohy najdete pouze v Materiálech na https://gitlab.fel.cvut.cz/rehakj16/brian-prirucka-assets.