

1. ÚLOHA – ZJIŠTĚNÍ RYCHLOSTI ROBOTA

Cílem této úlohy je zjistit na připravené dráze maximální rychlost přímočarého rovnoměrného pohybu robota. V platformě MakeCode se jeho rychlost udává v procentech, což je z fyzikálního hlediska nepřesné. Abychom později mohli pracovat s konkrétními hodnotami, budeme potřebovat znát jeho okamžitou rychlost v .

Pomůcky:

- dráha (runway)
- vhodné měřidlo délky
- stopky

Doporučený postup:

- 1) Zjistěte, jak dlouhý úsek je vyměřený na dráze.
- 2) Rozmyslete si, jaký bod na robotu budete sledovat (pro přesné určení času).
- 3) Připravte jednoduchý kód pro pohyb pomocí procent v prostředí MakeCode.
- 4) Nastavte, jak dlouho robot pojedje touto rychlostí, abyste ho nemuseli dohánět po učebně nebo chodbě.
- 5) Zjistěte rychlost i při nastavení nižší rychlosti a porovnejte výsledky, zda zjištěné údaje jsou rovnoměrné, případně měřte daný úsek znova.

Otázky k zamyšlení:

- 1) Jakým vzorečkem zjistíte rychlost tělesa při rovnoměrném přímočarém pohybu? (Můžete si pomoci jednotkou rychlosti)
- 2) Kolik procent nastavíte pro motory robota, když chcete zjistit poloviční nebo čtvrtinovou rychlost robota?

2. ÚLOHA – PLYNULÝ ROZJEZD ROBOTA PŘI STÁLÉM ZRYCHLENÍ

Jelikož již znáte, jakou rychlostí se robot pohybuje při zadání určitých procent pro motory, můžete začít pracovat se zrychlením za určitý čas.

Cílem této úlohy je nastavit pro robota rovnoměrné zrychlení z nulové rychlosti v_0 na maximální rychlost v_1 za jednu sekundu.

Doporučený postup:

- 1) Připravte si proměnné, které budete potřebovat – *okamžitou rychlost, okamžitý čas, celkový čas, zrychlení, počáteční rychlost (v_0), výslednou rychlost (v_1), proměnnou pro rychlost v procentech.*
- 2) Nastavte příkaz, aby se proměnné ukazovaly na kostce a mohli jste tak mít přehled o správnosti kódu.
- 3) Pomocí matematických operací nechte robota vypočítat *zrychlení*.
- 4) Nastavte proměnnou *okamžitou rychlost* pomocí proměnné *zrychlení* a *okamžitého času*.
- 5) Nastavte proměnnou *rychlost v procentech* pomocí *okamžité rychlosti*.
- 6) Využijte příkaz pro běh motorů za určitý čas a nastavte co nejmenší čas pro běh motoru – měňte proměnnou *okamžitý čas* o tu samou hodnotu, dokud nedosáhne stejného čísla jako *celkový čas*.

Otázky k zamyšlení:

- 1) Jakým vzorečkem zjistíte rovnoměrné *zrychlení* tělesa při přímočarém pohybu?
- 2) Jak byste vyjádřili *okamžitou rychlost*, když znáte vzoreček pro *zrychlení*?
- 3) Jak se dá zjistit *okamžitá rychlost v procentech*, když znáte maximální rychlost, tedy rychlost 100 %?

3. ÚLOHA – ROVNOMĚRNÉ ZRYCHLENÍ A ZPOMALENÍ PŘI DETEKCI BAREV

Váš robot již umí zrychlit z nulové na maximální rychlost. V této úloze bude robot rozpoznávat barvy pomocí senzoru barev a bude podle nich regulovat rychlost. Při detekci zelené barvy robot pojedí/zrychlí na maximální rychlost, při detekci žluté barvy robot zpomalí např. z 0,5 m/s (okamžitá rychlost) na 0,2 m/s a při detekci červené barvy robot plynule zastaví. Pro všechny změny rychlostí nastavte čas jednu sekundu. Detekci červené barvy programujte v JavaScriptu.

Abyste mohli začít programovat, přidejte své stavebnici senzor barev do takové pozice, aby byl senzor ve výšce připravených značek.

Cílem této úlohy je seznámit se s funkcí senzoru barev, umět s ním pracovat v kódu. Dále na základě zadání umět upravit nastavení proměnných, matematických operací a mít správný sled příkazů. V poslední části tohoto úlohy je cílem pochopení zápisu v textovém kódu JavaScript.

Vytvořené programy průběžně zkoušejte jak na simulátoru robota, tak i na připravené dráze.



1. Zrychlení při detekci zelené barvy

Doporučený postup:

- 1) Změřte výšku stojanu s barvou, případně ke stojanu přiložte robota, abyste věděli, do jaké výšky musíte senzor připevnit. Vhodné je volit místo v přední části robota a z boku. Připevnění volte staticky, aby se při jízdě senzor neuvolňoval nebo dokonce nepadal.
- 2) V prostřední MakeCode, v záložce s příkazy k senzorům, vyhledejte vhodnou vstupní událost pro detekci barvy a zvolte zelenou barvu.
- 3) Zkopírujte kód pro jízdu robota ze startu.
- 4) Zkontrolujte chod programu po detekci zelené barvy, upravujte části, které se mají lišit na rozdíl od původního kódu při zrychlení ze startu.
- 5) Pro plynulý chod robota při detekci barvy nastavte, aby po zaznamenání barvy pokračoval původní rychlostí další alespoň 0,2 s.

Otázky k zamyšlení:

- 1) Dle čeho určujeme číslo portu pro senzor v MakeCode?
- 2) Pokud má robot zrychlit při detekci zelené barvy, bude jeho počáteční rychlost nulová, když k té značce musel přijet, tedy se již pohyboval? Kde je uložena rychlost, kterou se robot aktuálně pohybuje?

2. Zpomalení při detekci žluté barvy

Doporučený postup:

- 1) Zkopírujte kód pro detekci zelené barvy a upravte ve vstupní událost (barvu).
- 2) Vytvořte novou proměnnou *zpomalení* a nastavte její hodnotu pomocí proměnných v_0 , v_1 a t .
- 3) Zkontrolujte chod programu po detekci žluté, upravujte části, které se mají lišit.
- 4) Kontrolujte výstupní hodnoty proměnných a chod motorů na simulátoru robota.
- 5) V případě chybných hodnot na simulátoru pište průběh cyklu na papír, jak robot s hodnotami pracuje. Upravujte váš postup tak, aby konečná rychlost v m/s byla správná.
- 6) Pro plynulý chod robota při detekci barvy nastavte, aby po zaznamenání barvy pokračoval původní rychlostí alespoň další 0,2 s.

Otázky k zamyšlení:

- 1) Jak vyjádříme zpomalení, aby výsledná hodnota nebyla záporná? Která rychlost je na počátku větší?

3. Zastavení při detekci červené barvy

Překlikněte do textového kódu (JavaScript).

Doporučený postup:

- 1) Zkopírujte kód pro detekci žluté barvy a upravte ve vstupní událost (barvu).
- 2) Projděte příkaz za příkazem a ve vaší skupince vyhodnocujte, zda je příkaz vhodný pro zastavení, nebo zda ho musíte změnit či odebrat.
- 3) Průběžně kontrolujte změny i blokovém editoru a na simulátoru robota.

Otázky k zamyšlení:

- 1) Jaká musí být konečná rychlost robota, když má zastavit?
- 2) Bude na konci programu (po ukončení cyklu s podmínkou) příkaz pro běh motorů, když má robot zastavit?

4. ÚLOHA – ODBOČENÍ NA KŘIŽOVATCE

V rámci této úlohy musí robot zaznamenat křižovatku pomocí modré značky. Po zaznamenání modré barvy robot zpomalí a otočí se o 90° doprava. Po odbočení plynule zvýší rychlost na maximální hodnotu a po dvou sekundách zastaví.



Doporučený postup:

- 1) Připravte vstupní událost pro detekci modré barvy.
- 2) Vložte kód pro běh každého motoru zvlášť tak, aby se současně točily oba, ale každý jinou rychlostí.
- 3) Zkoušejte otáčení na připravené dráze a hodnoty pro motory dle chování robota upravujte.
- 4) Doplněte kódy pro zpomalení a zrychlení, správně kód začleňte do připraveného kódu při detekci modré barvy.
- 5) Upravte čas zpomalení tak, aby se robot při jízdě nedostal příliš do křižovatky a neodbočil tím pádem do špatného pruhu.

Otázky k zamyšlení:

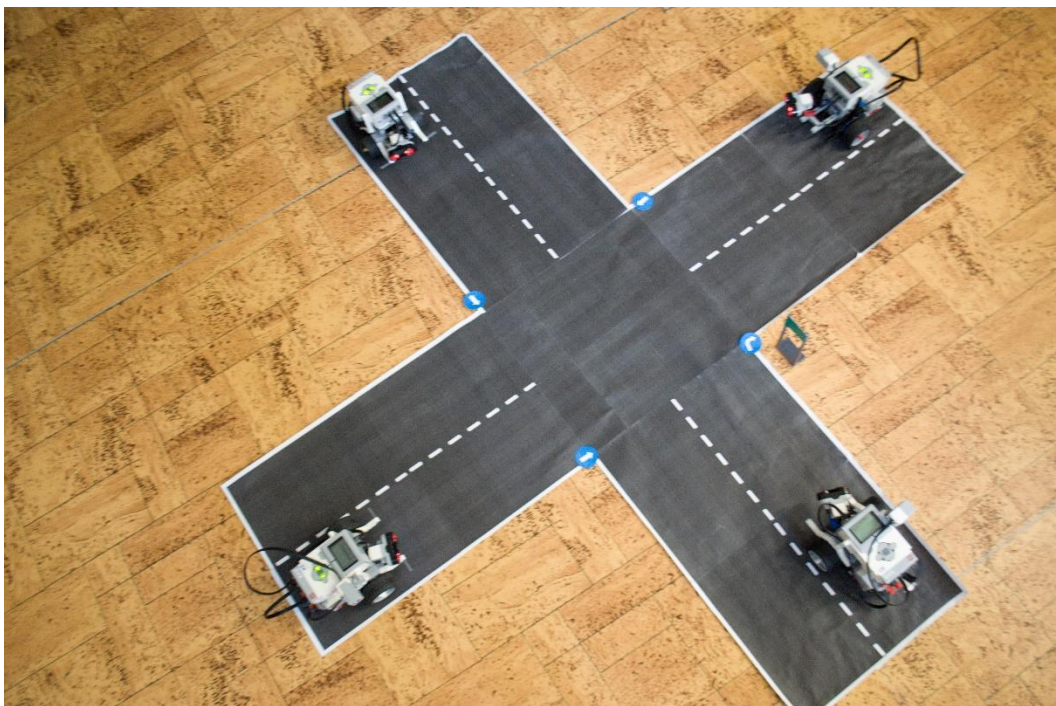
- 1) Který motor se musí otáčet rychleji a který pomaleji, pokud robot zahýbá doprava?

5. ÚLOHA – DEJ PŘEDNOST V JÍZDĚ

Pro tuto úlohu je opět jako dráha připravena křižovatka, přes kterou tři roboti pokračují rovně, jeden robot odbočí vpravo (využijte kód z předchozí úlohy).

Úloha je koncipována tak, že robot přijede na neoznačenou křižovatku a musí hlídat její pravou část, jelikož platí pravidlo pravé ruky, tedy přednost v jízdě zprava.

Při startu se robot plynule rozjede na maximální rychlost (případně trochu nižší), pokud zaznamená překážku, která je blíž jak 50 cm, robot musí zpomalit (stejná rychlost jako při detekci žluté barvy z úlohy s barevnými značkami). Přiblíží-li se robot k objektu/překážce na vzdálenost menší než 35 cm, musí plynule zastavit. Pokud překážka po pravé straně odjede, robot se musí opět rozjet do maximální rychlosti (v tomto případě nastavte pohyb na 6 sekund), případně rychlosti nastavené pro zpomalení, jestliže je ve vzdálenosti do 50 cm jiný objekt.



Doporučený postup:

- 1) Zjistěte, jaký senzor je vhodný na detekci překážky.
- 2) Připevněte senzor pevně k robotu tak, aby byl nakloněn mírně doprava a dokázal tak detekovat překážku v pravé části křižovatky. (Doporučuji naklonění vyzkoušet na dráze.)
- 3) Vytvořte jednoduchý program, který po detekci objektu vypíše na displeji kostky danou vzdálenost překážky.
- 4) Vytvořte proměnnou *vzdálenost*, do které se bude ukládat hodnota ze senzoru.

- 5) Upravte program tak, aby při detekci překážky ve vzdálenosti do 50 cm začal zpomalovat – využijte kód z předchozích úloh.
- 6) Přejděte do JavaScriptové části a doplňte kód pro zastavení tak, aby robot při detekci ve vzdálenosti menší než 35 cm zastavil.
- 7) Doplňte zbývající kód pro zrychlení v případě, že před robotem překážka není – zrychlení na maximální rychlost.
- 8) Průběžně testujte na simulátoru a na vašem robotu, případně kód včetně vzdáleností upravte tak, aby robot nezastavil daleko před křižovatkou nebo v křižovatce.

Otázky k zamyšlení:

- 1) Co za hodnoty zpracovává senzor, který potřebujete?
- 2) V jakém portu máte připojený senzor a jaký port máte nastavený v MakeCode?
- 3) Jak vyjádříte kódem větu „Když je vzdálenost menší než 50 cm, zpomal na 0,2 m/s“