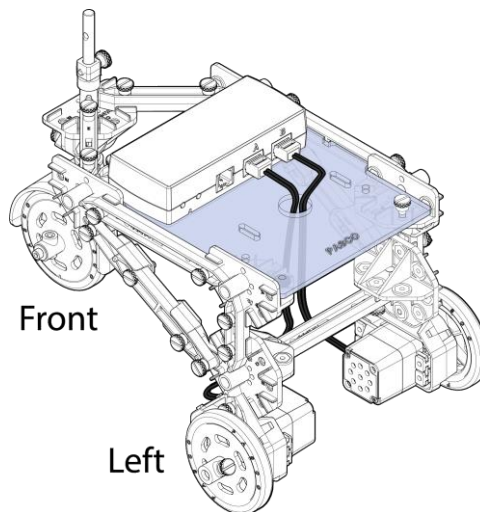

Otočte StructureBOT

Cíle

- Naučte se, jak přimět StructureBOT, aby se otočil doleva a doprava o 90° a otočil se kolem středu BOTu.

Materiály a vybavení

Číslo dílu	Popis	mn ožs tví
ME-7039	StructureBOT, sestaveno	1
PS-3232	//control.Node	1
	Software pro sběr dat PASCO Capstone™	



Obrázek 1. StructureBOT sestavený s //control.Node

Postup

Založit

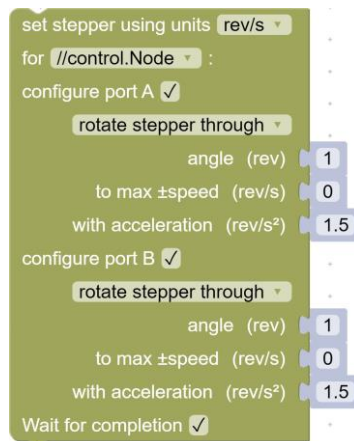
1. Sestavte StructureBOT, jak je popsáno v „Sestavení StructureBOT s pohonem 2 kol“.
2. Nabijte //control.Node.
3. Ujistěte se, že je krokový motor na levé straně BOT zapojen do portu A //control.Node a že krokový motor na pravé straně BOT je zapojen do portu B.
4. Připojte //control.Node k softwaru PASCO Capstone přes Bluetooth.

POZNÁMKA: Pokyny týkající se softwarových úloh, jako je připojení bezdrátových zařízení a používání Blockly, naleznete v nabídce Náповěda v PASCO Capstone.

5. Vytvořte graf úhlu, Port A versus čas. Poté přidejte plochu plotu a vykreslete úhel, Port B versus čas.

Část 1: Zatáčka vlevo

1. V pracovním prostoru Blockly Code začněte tímto blokem:



Zpočátku tento kód nedělá nic, protože hodnoty max ±rychlosti (ot./s) jsou 0.

2. Pro každý krokový motor zadejte experimentální kladné a záporné hodnoty pro maximální ±rychlost (otáčky/s). Prozkoumejte znaménko potřebné pro každý krokový motor, aby se BOT

POZNÁMKA: Možné hodnoty se pohybují od -2 ot/s do +2 ot/s.

otáčel (točil kolem své svislé osy).

3. Při úhlu 1 otáčky pro oba steppery, pod jakým úhlem (přibližně) se BOT otáčí?

4. O jaký úhel se musí otočit kola, aby se BOT otočil o 90°?

s = délka oblouku, W = vzdálenost mezi hnacími koly, D = průměr kola, N = počet otáček kola (nebo hřídele motoru)

délka oblouku = (poloměr) (úhel v radiánech)

$$s = (W/2)(\pi/2) = \frac{W\pi}{4} \quad (1)$$

Délka oblouku může být také vyjádřena jako vzdálenost, kterou kolo urazí.

délka oblouku = (obvod kola) (počet otáček)

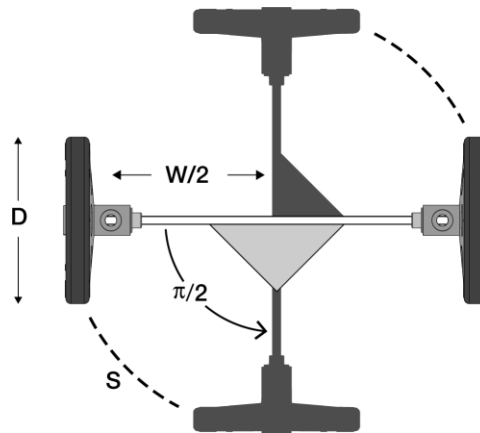
$$s = \pi DN \quad (2)$$

Nastavte rovnici (1) rovnou rovnici (2).

$$\pi DN = \frac{W\pi}{4}$$

Vyřešte počet otáček kola.

$$N = \frac{W}{4D} \quad (3)$$



5. Abychom mohli odbočit doleva, musíme nejprve BOT zastavit (v případě, že se BOT již pohybuje). Vytvořte funkci s názvem turnLeft. Vytvořte následující proměnné:

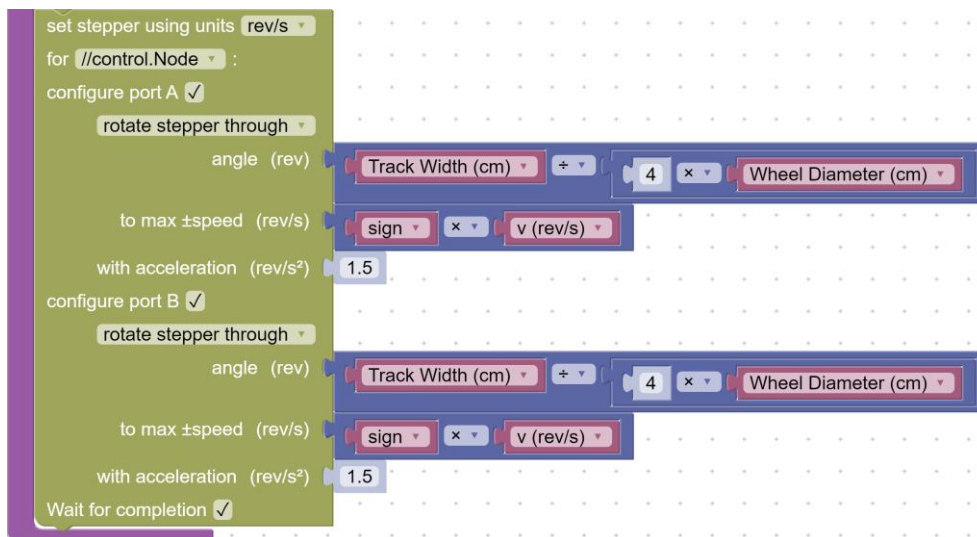
- Průměr kola (cm)
- Šířka stopy (cm)
- v (ot./s)
- podepsat

Uvnitř funkce nastavte tyto proměnné na hodnoty uvedené níže. Zahrňte krokový blok, který zastaví oba krokové motory.

```

to turnLeft
  set sign to 6.38
  set Track Width (cm) to 17.57
  set v (rev/s) to 1.5
  set sign to -1
  set stepper using units rev/s
  for //control.Node :
    configure port A ✓
    stop
    with acceleration (rev/s²) 1.5
    configure port B ✓
    stop
    with acceleration (rev/s²) 1.5
  Wait for completion ✓
  
```

6. Přidejte další krokový blok k funkci, díky které se BOT otočí doleva o 90°.



Část 2: Kalibrace šířky stopy

Aby se BOT otočil přesně o 90°, musí být přesně změřena šířka stopy. Jedním ze způsobů, jak přesně určit šířku stopy, je provést osm zatáček o 90 stupňů a zjistit, zda se BOT po provedení těchto dvou úplných kruhů vrátí do výchozího bodu.

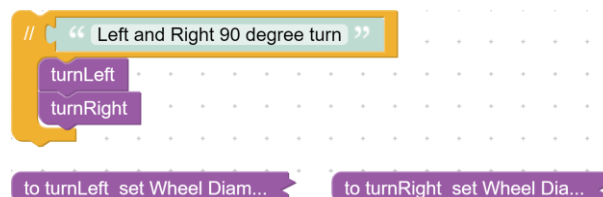
1. Zarovnejte BOT s čarou na podlaze.
2. V kódu použijte funkci turnLeft osmkrát za sebou. Končí BOT na lince? Pokud ne, změňte hodnotu šířky stopy v kódu a otestujte ji znovu.

Část 3: Odbočte doprava

1. Vytvořte funkci s názvem turnRight. Jaká je jediná změna, kterou musíte udělat, když odbočíte doleva kódu, aby BOT odbočil doprava?
2. Použijte své funkce, které se několikrát opakují, aby se robot otočil doprava nebo doleva.

Část 4: Uložte funkci

1. Vytvořte blok Notes s vysvětlením, že tato funkce způsobí, že se BOT otočí o 90° doleva nebo doprava.
2. Do tohoto bloku Notes vložte funkce turnLeft a turnRight.
3. Klepněte pravým tlačítkem myši na funkce a sbalte je.



4. Uložte funkci jako Turn-Left-Right.pcbx.