

Plano de Aula – À Distância

Tema:

Utilização do sensor de distância para navegação autônoma e prevenção de colisões com o robô VEX EXP

Duração da aula:

1 aula de 50 minutos

Componentes curriculares envolvidos:

- Ciências
 - Tecnologia
 - Matemática
 - Pensamento Computacional
 - Educação Digital
-

Turmas indicadas:

4º ao 5º ano do Ensino Fundamental

Objetivos da aula:

- Utilizar sensores de distância para controlar o movimento do robô.
 - Compreender como a leitura de sensores influencia decisões em tempo real.
 - Escrever e ajustar códigos para tornar o robô mais autônomo.
 - Desenvolver lógica de programação aplicada a situações reais de obstáculos.
-

Competências da BNCC:

- Competência 1: Conhecimento

- Competência 2: Pensamento científico, crítico e criativo
 - Competência 5: Cultura digital
 - Competência 6: Trabalho e projeto de vida
 - Competência 7: Argumentação
-

Habilidades da BNCC:

- (EF04CI02) Observar e compreender o funcionamento de sensores.
 - (EF05MA20) Utilizar lógica e variáveis em resolução de problemas.
 - (EF04EM04) Criar códigos com base em condições (if/while).
 - (EF05CI06) Aplicar conhecimentos científicos para solucionar problemas práticos.
-

Materiais necessários:

- Kit VEX EXP com sensor de distância
 - VEXcode EXP (modo Python ou Blocks)
 - Dispositivo com VEXcode instalado
 - Buckyballs ou objetos para representar obstáculos
 - Campo 3x3 ou espaço livre na sala
-

Etapas e Desenvolvimento da Aula (Passo a Passo):

1. Introdução ao sensor de distância (5 min)

- Apresente o sensor de distância e como ele “vê” objetos por meio de leituras numéricas.
- Explique: quanto menor o número, mais próximo o obstáculo.

2. Montagem e configuração (10 min)

- Construa o BaseBot e conecte o sensor à frente do robô.
- Abra o VEXcode EXP, selecione o modelo BaseBot (Drivetrain 2-motor) e configure o sensor.
- Posicione obstáculos no campo com Buckyballs ou materiais disponíveis.

3. Criação do código básico (10 min)

- Desenvolva um código com estrutura de repetição que pare o robô ao se aproximar de um objeto.
- Use estrutura `while not (distância < X) e drive.forward()`.

4. Testes com obstáculos (15 min)

- Faça o robô se aproximar de uma Buckyball e parar automaticamente.
- Adicione comandos para dirigir até uma segunda Buckyball.
- Teste diferentes distâncias e repositone os obstáculos para verificar reações do robô.

5. Ajustes e reflexão (5 min)

- Pergunte:
 - “Qual distância funcionou melhor?”
 - “O robô parou no tempo certo?”
 - “Como melhorar o código?”
-

Subindo de Nível:

- Adicione mais obstáculos ou paredes.
 - Faça o robô percorrer uma rota com curvas usando `turn_for`.
 - Imprima os valores do sensor na tela do robô para análise.
-

Conteúdos trabalhados:

- Sensores e suas aplicações
 - Condicionais e repetição na programação
 - Testes e ajustes em projetos robóticos
 - Tomada de decisão automatizada
-

Dicas para o professor:

- Use fita para marcar distâncias no chão.
 - Estimule testes comparativos entre valores diferentes.
 - Combine com desafios de precisão de parada.
-

Discussões e conclusões:

- Como o robô “sabe” onde parar?
 - O que acontece se mudarmos o valor do sensor?
 - Quais situações do mundo real usam sensores assim?
-

Interdisciplinaridade:

- Ciências: percepção, distância, sensores
 - Matemática: medição e comparação
 - Tecnologia: lógica de programação
 - Educação Digital: leitura de dados e automação
-

Avaliação formativa:

- Compreensão do funcionamento do sensor
 - Aplicação correta dos blocos de código
 - Clareza ao justificar ajustes no código
 - Cooperação durante os testes e revisões
-

Resultados esperados:

- Compreensão da leitura e uso de sensores
- Criação de código com condicionais e ajustes
- Autonomia no controle do robô
- Desenvolvimento de lógica prática e colaborativa