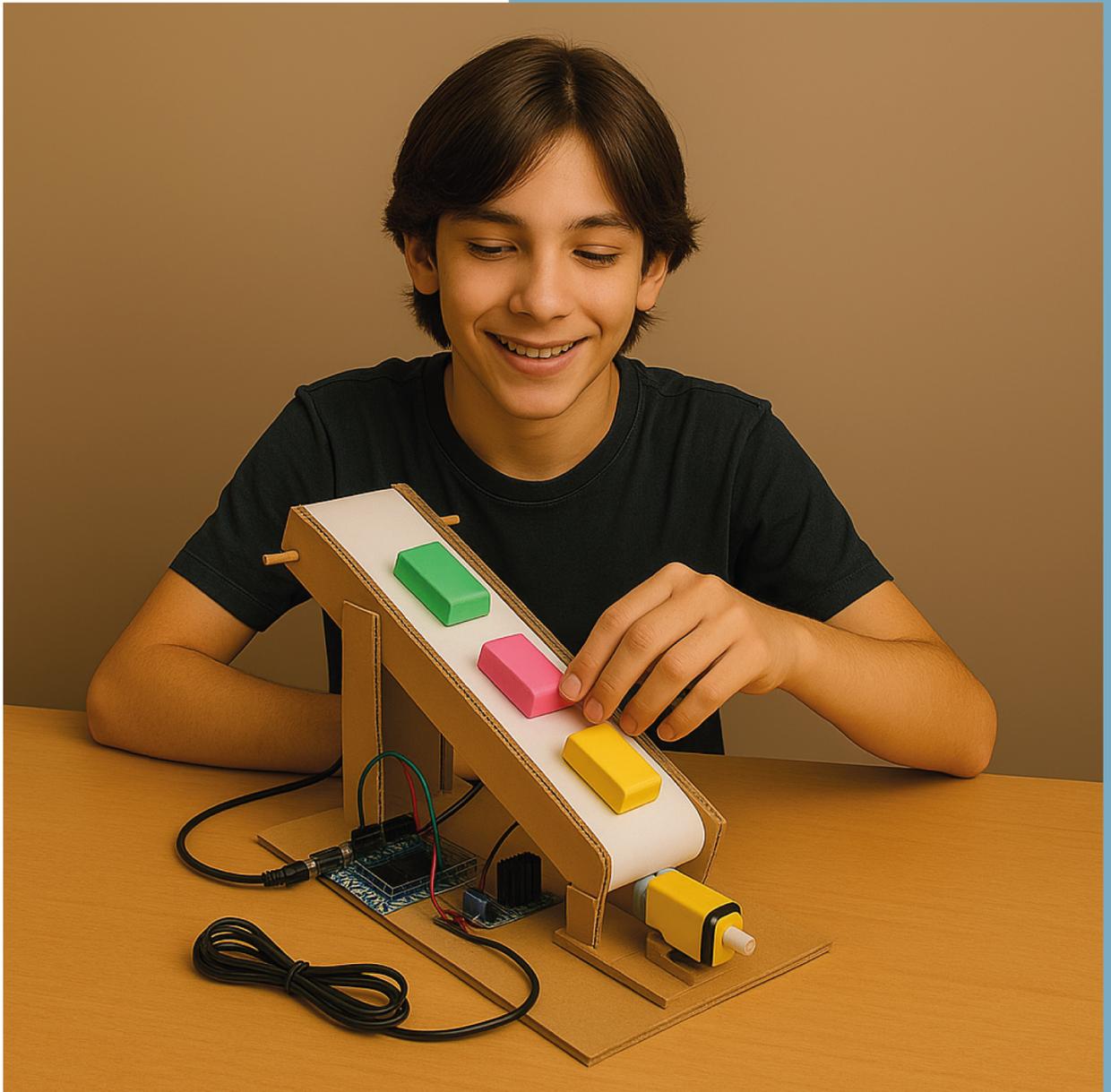


ROBÓTICA

AULA 37

Primeiros Passos Módulo 3



Esteira rolante - II



Diretoria de Tecnologia e Inovação

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Adilson Carlos Batista

Validação de Conteúdo

Viviane Dziubate Pittner

Darice Alessandra Deckmann Zanardini

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

2024

SUMÁRIO

Introdução	2
Objetivos desta aula	2
Roteiro da aula	3
1. Contextualização	3
2. Conteúdo	4
3. Montagem e programação	9
4. Feedback e finalização	21
Referências bibliográficas	21

Introdução

Você entrou em contato com o tema esteira rolante ou esteira transportadora na aula anterior e pôde verificar que elas são muito importantes nos diversos setores industriais e logísticos. Além da compreensão do tema, usando papelão e outros materiais, você construiu a estrutura do protótipo de uma esteira rolante que utilizaremos nesta aula para inserir a parte eletrônica do projeto e fazer a programação.

Objetivos desta aula

- Montar a parte eletrônica do protótipo;
- Programar o protótipo da esteira rolante.

Lista de materiais

- Arduino Uno R3;
- Motor DC com caixa de redução;
- Ponte H;
- Fonte chaveada 9V;
- Sensor IR;
- Jumpers macho-macho e macho-fêmea.



Roteiro da aula

1. Contextualização

Na aula anterior, você aprendeu que as esteiras rolantes são sistemas altamente adaptáveis e eficazes, que integram conhecimentos de física e engenharia para otimizar fluxos de trabalho em diversos setores. Ao automatizar o transporte de materiais, elas proporcionam um aumento significativo na produtividade, redução de custos operacionais e melhoria na qualidade final dos produtos.

A mecânica clássica explica o funcionamento básico das esteiras, enquanto a energia elétrica, transformada em energia mecânica por motores, impulsiona o movimento. A escolha de materiais resistentes e o design adequado garantem a durabilidade e o desempenho desses equipamentos. Controladores eletrônicos permitem ajustar a velocidade e a direção das esteiras, oferecendo flexibilidade e segurança.

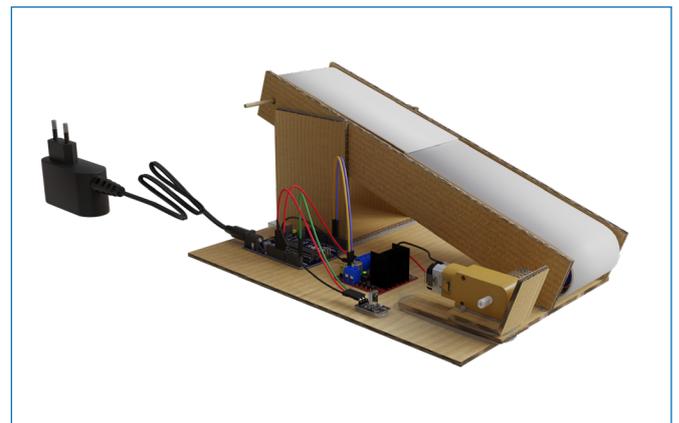
Como visto, existem diversos tipos de esteiras, cada qual adaptado a diferentes aplicações e materiais. Desde esteiras de roletes livres para transporte de insumos até esteiras de manta para materiais a granel, a escolha depende das necessidades específicas de cada indústria ou comércio.

Em resumo, as esteiras rolantes são fundamentais para a automação industrial,

otimizando processos, reduzindo custos e contribuindo para um ambiente de trabalho mais seguro e eficiente.

Você construiu toda a estrutura da esteira rolante, agora chegou o momento de finalizar inserindo a parte eletrônica e fazer a programação de funcionamento do protótipo. Com a parte eletrônica, ela deverá ficar conforme Figura 1.

Figura 1 – Protótipo de esteira rolante



Fonte: Roberto Carlos Rodrigues, 2024.

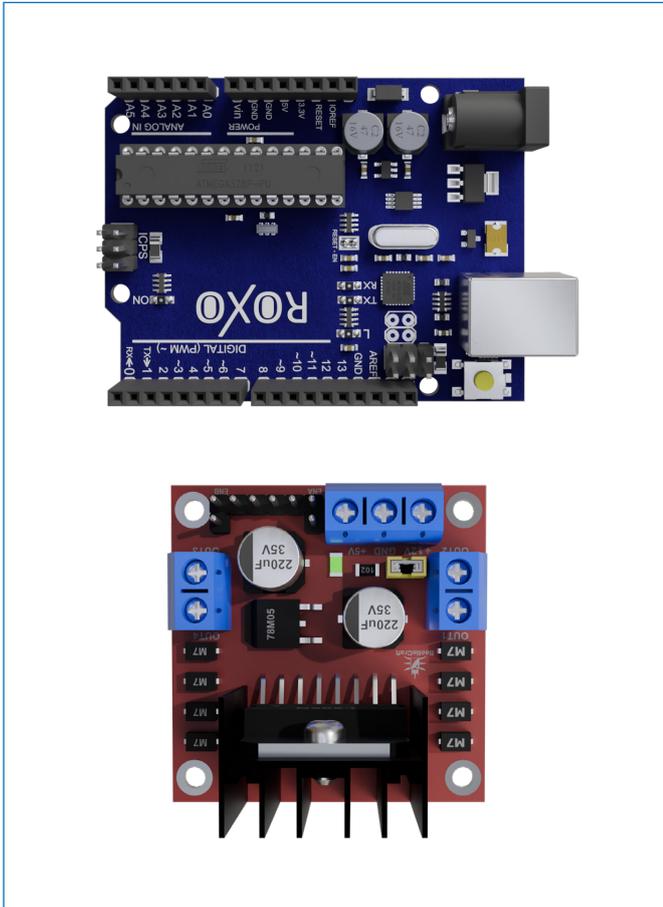
Antes de partirmos para a segunda parte da montagem, vamos rever alguns componentes que serão utilizados. Lembrando que você teve contatos com eles em diferentes aulas.

2. Conteúdo:

Os primeiros componentes que serão destacados são: o Arduino e a ponte H.

O ARDUINO E A PONTE H: UMA DUPLA DINÂMICA!

Figura 2 – Arduino e ponte H



Fonte: Roberto Carlos Rodrigues, 2024.

Podemos imaginar o **Arduino** como um pequeno computador que você pode programar para fazer diversas coisas! Ele é como um Lego eletrônico, onde você pode conectar vários componentes e criar

seus próprios projetos, ou seja, possibilita dar vida a suas ideias!

Com ele, você pode:

Controlar motores: fazer um carrinho se mover, um braço robótico se mexer, etc.

Ler sensores: detectar movimento, medir temperatura, medir distância, etc.

Acender LEDs: criar efeitos de luz, sinais luminosos e até mesmo displays.

Criar interfaces: construir botões, interruptores e outros elementos para interagir com seus projetos.

A **ponte H** é como um superpoder para o Arduino. Ela permite que o Arduino controle motores que consomem muita energia, como os motores de um carrinho ou de um braço robótico. Sem a ponte H, o Arduino não teria força suficiente para ligar esses motores.

O Arduino envia comandos para a ponte H, dizendo se o motor deve girar para frente, para trás ou parar, ela, por sua vez, liga e desliga a energia do motor conforme os comandos do Arduino. É como se o Arduino fosse o cérebro e a ponte H fosse os músculos do seu projeto!

Mas como a ponte H funciona?

A ponte H L298N é usada com o Arduino para controlar motores DC, permitindo o ajuste da direção e da velocidade. É especialmente útil em projetos de robótica, como carrinhos e braços robóticos.

Principais terminais da ponte H:

12V: entrada de energia para os motores. Conecte aqui uma fonte externa se necessário, especialmente para motores mais potentes.

GND: terra, necessário para definir a referência de corrente, conectando-o ao GND do Arduino.

5V: fornece tensão de controle para a ponte H e pode alimentar o Arduino em alguns casos, mas, geralmente, o Arduino deve ser alimentado separadamente para maior estabilidade.

Controle de velocidade com PWM e jumpers ENA e ENB:

Os jumpers **ENA** e **ENB** controlam o estado dos motores, eles são como chaves que ligam e desligam os motores.

Para controle de velocidade, utilize o **PWM (Modulação por Largura de Pulso)** no pino IN1 e conectado ao Arduino. O PWM permite variar a energia fornecida ao motor, ajustando a velocidade de forma suave. É como se a gente estivesse piscando a luz muito rápido, só que em vez de luz, a gente está ligando e desligando a energia do motor. Quanto mais tempo a luz ficar ligada, mais rápido o motor gira!

Fonte de alimentação externa:

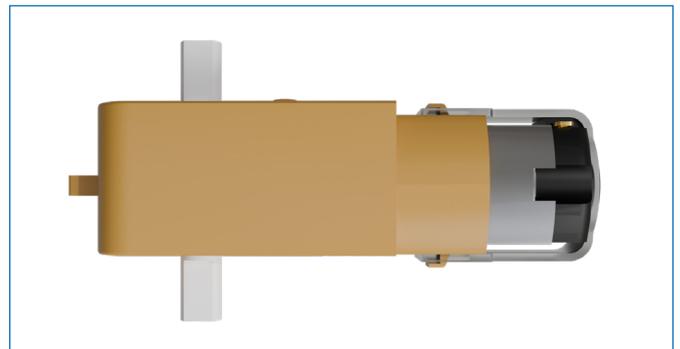
Para projetos que utilizam motores de alta potência, uma fonte externa dedicada é a opção mais segura e eficiente. Conec-

te o positivo da fonte ao terminal 12V da ponte H e o negativo ao GND da ponte H e ao GND do Arduino.

Utilizar uma fonte externa ajuda a manter a estabilidade do Arduino e evita sobrecargas. Lembre-se, sempre verifique os limites de corrente e tensão antes de utilizar o pino VIN do Arduino. Antes de ligar a alimentação, revise todas as conexões para evitar danos à ponte H e ao Arduino. Mais detalhes sobre a utilização da ponte H podem ser encontrados na **Aula 27 - Elevador Elétrico II**. Com a ponte H e o Arduino, você pode construir carrinhos, robôs e muitas outras coisas incríveis!

E aí, gostou de relembrar o uso desses dois componentes? Juntos, eles formam uma dupla imbatível para criar projetos incríveis! Vamos para o próximo **componente!**

Figura 3 – Motor DC com caixa de redução



Fonte: Roberto Carlos Rodrigues, 2024.

Imagine um motorzinho superforte que você pode controlar! Esse motorzinho é chamado de motor DC com caixa

de redução. Ele é como um músculo para os seus projetos de Robótica!

Ele funciona da seguinte forma: quando você liga o motor em uma bateria, ele começa a girar. A parte legal é que ele tem uma caixa de redução que faz ele girar mais devagar, mas com mais força! É como se você trocasse uma bicicleta comum por uma bicicleta com marchas: você pode pedalar mais devagar e subir morros mais altos.

Para que serve?

Esse motor é perfeito para construir:

Carrinhos: faça um carrinho super-rápido e que consegue subir rampas!

Braços robóticos: crie um braço robótico que pode pegar objetos com muita força.

Miniaturas: faça miniaturas que se mexem, como um trem ou um elevador.

E muito mais!

O que você precisa saber sobre ele?

Tensão: ele funciona com baterias de 3 a 6 volts. Quanto maior a tensão, mais rápido ele gira e mais forte ele fica.

Torque: essa é a força que o motor faz para girar. Quanto maior o torque, mais pesado o objeto que ele consegue mover.

Redução: a caixa de redução faz o motor girar mais devagar, mas com mais força.

Tamanho: ele é bem pequeno e leve, perfeito para seus projetos.

Com esse motor, suas ideias vão virar realidade!

Vamos para o receptor IR!

Figura 5 – Receptor IR



Fonte: Roberto Carlos Rodrigues, 2024.

Imagine um controle remoto de TV. Quando você aperta um botão, ele envia um sinal invisível para a TV, certo? Esse sinal invisível é feito de luz infravermelha, e o componente da TV que recebe esse sinal é chamado de receptor IR. Ele é como um ouvidinho eletrônico que capta a luz infravermelha. Quando você aponta o controle remoto para a TV e aperta um botão, o receptor IR dentro da TV detecta essa luz e a transforma em um sinal elétrico. Esse sinal elétrico, por sua vez, diz à TV o que você quer fazer, como aumentar o volume ou mudar de canal.

E em uma esteira rolante que é o nosso projeto?

Funciona da mesma forma! O controle remoto da esteira rolante envia sinais infravermelhos para um receptor IR instalado no protótipo. Quando o receptor capta esses sinais, ele “entende” qual botão você apertou e envia essa informação para o circuito eletrônico da esteira.

Por exemplo:

Se você apertar o botão “iniciar”, o receptor IR envia um sinal para a esteira começar a se mover.

Se você ajustar a velocidade, o receptor IR envia um sinal para a esteira mudar a velocidade.

Em resumo:

Receptor IR: é um componente eletrônico que capta a luz infravermelha.

Função na esteira rolante: recebe os sinais do controle remoto e “diz” à esteira o que fazer.

O receptor IR é uma peça fundamental para o funcionamento do controle remoto de uma esteira rolante, permitindo que você controle a esteira de forma simples e intuitiva.

Figura 5 – Fonte chaveada 9V



Fonte: Roberto Carlos Rodrigues, 2024.

Uma fonte chaveada de 9V é um dispositivo eletrônico que converte a tensão de entrada (geralmente AC da rede elétrica) em uma tensão de saída estável de 9 volts, utilizando uma técnica chamada de comutação ou chaveamento. Ao contrário das fontes lineares, que operam de forma contínua, uma fonte chaveada funciona ligando e desligando rapidamente a corrente elétrica (em alta frequência) por meio de transistores, o que a torna mais eficiente, compacta e leve.

Características principais de uma fonte chaveada de 9V:

Tensão de saída: 9V (DC - corrente contínua), com uma tolerância que pode variar dependendo da qualidade da fonte (geralmente $\pm 5\%$).

Tensão de entrada: modelos universais aceitam uma ampla gama de tensões de entrada (geralmente de 100V a 240V AC), permitindo o uso em diferentes regiões do mundo.

Corrente de saída: a capacidade de corrente varia conforme o modelo. Fontes comuns podem fornecer 500mA, 1A, 2A, ou mais, dependendo do que se deseja alimentar.

Potência nominal: a potência nominal é a multiplicação da tensão de saída pela corrente de saída ($P = V \cdot I$). Por exemplo, uma fonte de 9V e 1A teria uma potência de 9W.

Eficiência: fontes chaveadas têm alta eficiência, frequentemente entre 70% e 90%. Essa eficiência se deve ao uso da comutação rápida, que minimiza a dissipação de calor.

Regulação de tensão: uma boa fonte chaveada tem uma regulação de tensão eficiente, ou seja, a tensão de saída permanece estável mesmo que a tensão de entrada ou a carga variem.

Proteções:

Proteção contra sobrecorrente: impede que a fonte forneça corrente excessiva, o que poderia danificar tanto o próprio dispositivo quanto o equipamento alimentado.

Proteção contra sobretensão: garante que a tensão de saída não exceda o valor especificado.

Proteção contra curto-circuito: desliga a fonte em caso de curto-circuito na saída.

Proteção contra superaquecimento: desliga a fonte caso a temperatura interna atinja um valor perigoso.

Dimensões e peso: devido à maior eficiência energética, fontes chaveadas são menores e mais leves que as fontes lineares equivalentes.

Agora que você lembrou os componentes utilizados no protótipo, vamos para a montagem.



3. Montagem e programação

Com a estrutura do protótipo de esteira rolante finalizada na aula 36, cole o motor DC, conforme Figuras 6 e 7, e o sensor IR, conforme Figuras 8 e 9.

Figura 6 - Colagem da peça de suporte no motor DC

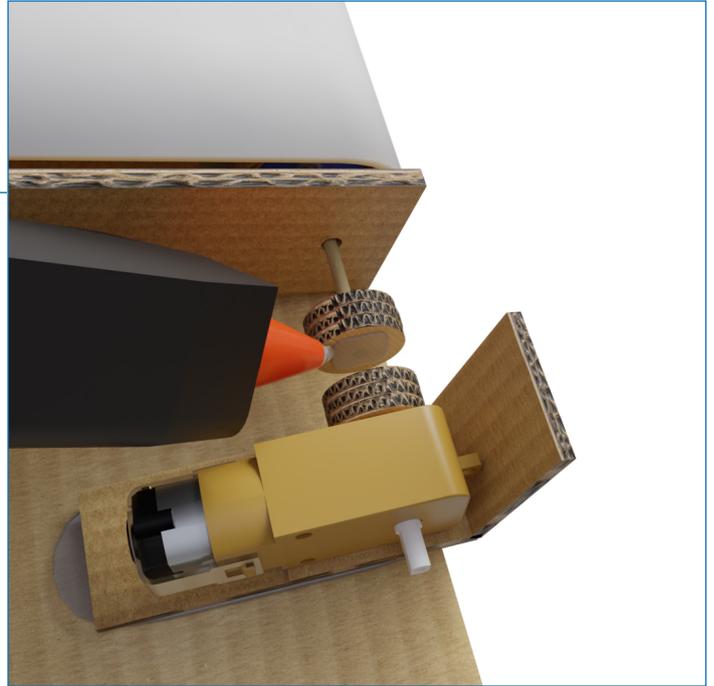


Figura 7 - Colagem da peça de suporte no motor DC

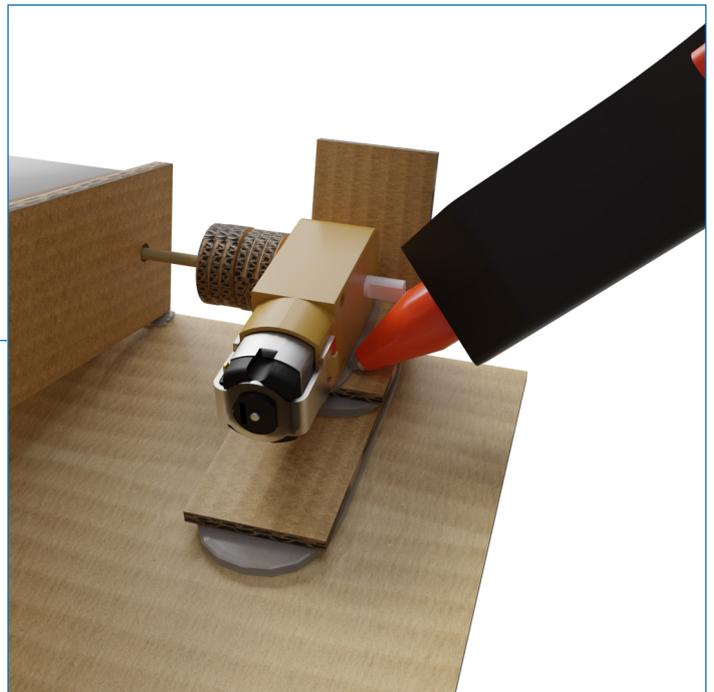


Figura 8 – Colagem do sensor IR no protótipo

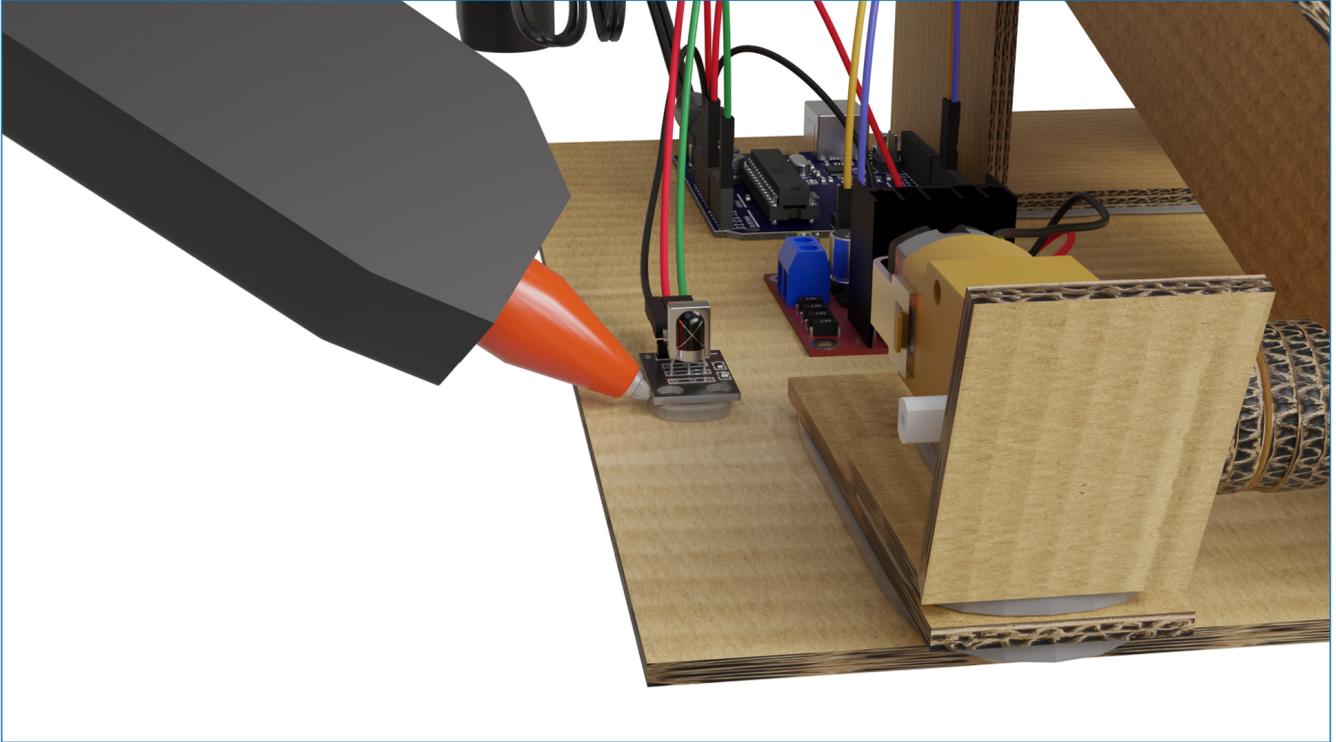
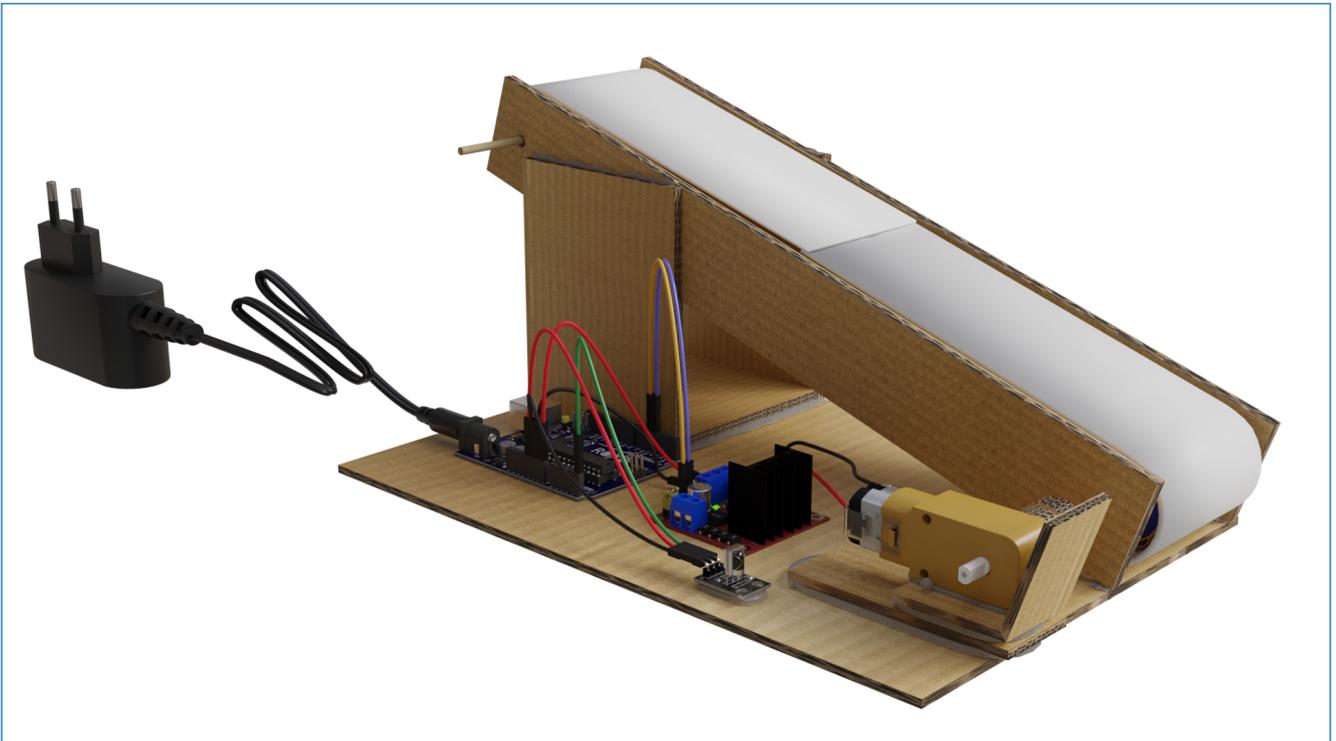


Figura 9 – Protótipo de esteira rolante pronto



Esteira rolante - II

Conecte a protoboard ao Arduino:

Utilize um jumper vermelho para conectar o pino 5V do Arduino ao trilho +5V da protoboard. Utilize um jumper azul para conectar o pino GND do Arduino ao trilho GND da protoboard.

Essas conexões fornecem a alimentação necessária para a protoboard e os componentes conectados a ela.

Conecte a ponte H ao Arduino e ao motor de redução:

Utilize um jumper vermelho para conectar o terminal 12V da ponte H ao pino VIN do Arduino ou a uma fonte externa para alimentação (o pino VIN do Arduino suporta fontes de até 12V, mas sempre confira as especificações do seu motor para evitar sobrecarga).

Ligue o GND da ponte H ao GND do Arduino para estabelecer uma referência comum. Conecte os terminais do motor DC aos pinos OUT1 e OUT2 da ponte H para definir o sentido de rotação (a inversão dos fios altera a direção de giro).

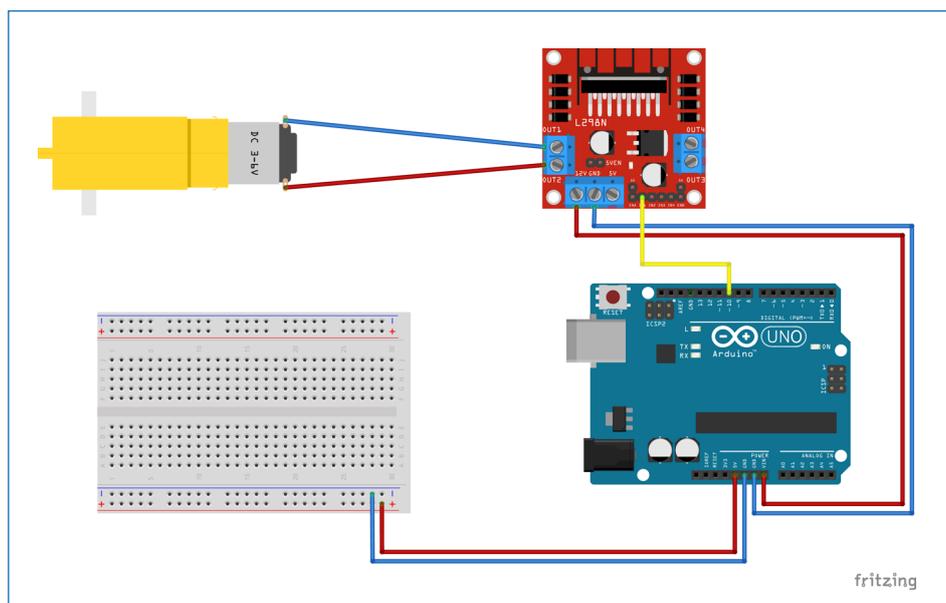
A polaridade dos fios do motor é importante para definir o sentido de rotação.

Conecte os pinos de controle da ponte H ao Arduino:

Por último, utilize um jumper macho-fêmea e conecte o pino IN1 da ponte H ao pino digital PWM 10 do Arduino. Se desejar inverter a direção do motor DC, conecte o pino IN2 da ponte H ao pino digital PWM, ao invés do IN1.

Importante: Certifique-se de que todas as outras conexões estejam corretas antes de ligar a fonte de alimentação.

Figura 10 - Conexão da ponte H e motor DC

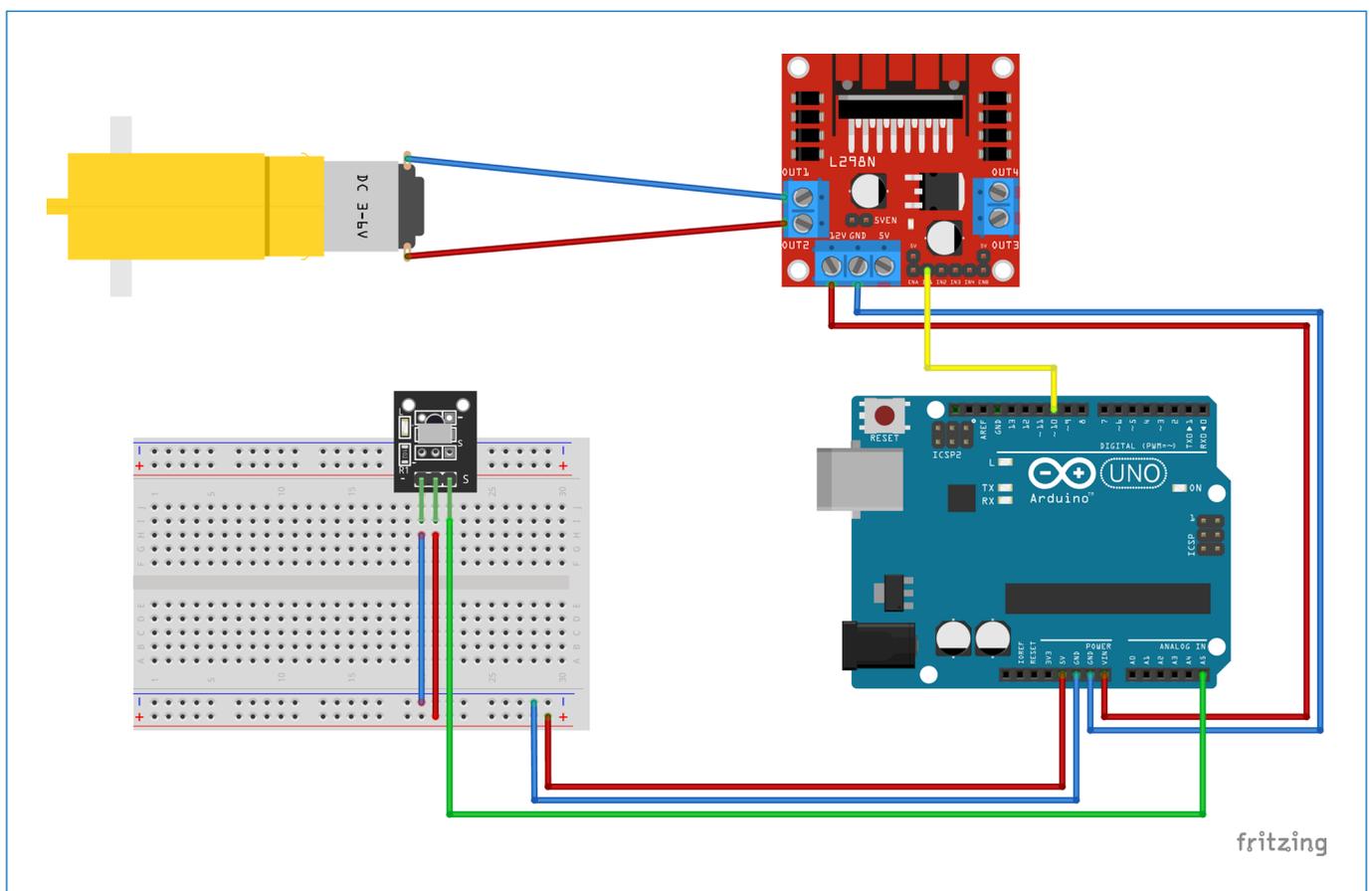


Esteira rolante - II

Agora, vamos conectar o sensor IR à protoboard.

Com um jumper vermelho, conecte o pino VCC do sensor IR ao trilho positivo da protoboard. Utilize um jumper azul para conectar o pino GND do sensor IR ao trilho negativo da protoboard. Com um jumper verde, conecte o pino de sinal (data) do sensor IR à porta analógica A5 do Arduino. O exemplo aqui é somente para você compreender a montagem, o sensor IR você poderá colar diretamente no protótipo e fazer as conexões sem a protoboard.

Figura 11 - Conexão do sensor IR na protoboard e ao Arduino



Verificações:

- Verifique se todas as conexões estão corretas e seguras.
- Certifique-se de que a polaridade dos componentes está correta.
- Antes de ligar a fonte de alimentação, faça uma última verificação visual para evitar curtos-circuitos.
- Para alimentação, utilize uma fonte chaveada 9V.

Programação

Para a programação, utilizaremos o controle remoto para a esteira, então teremos que programá-lo com os códigos para cada função do botão, usando a biblioteca <IRremote.hpp> visto que o mBlock não traz esses códigos. Importante salientar que cada controle tem a sua numeração.

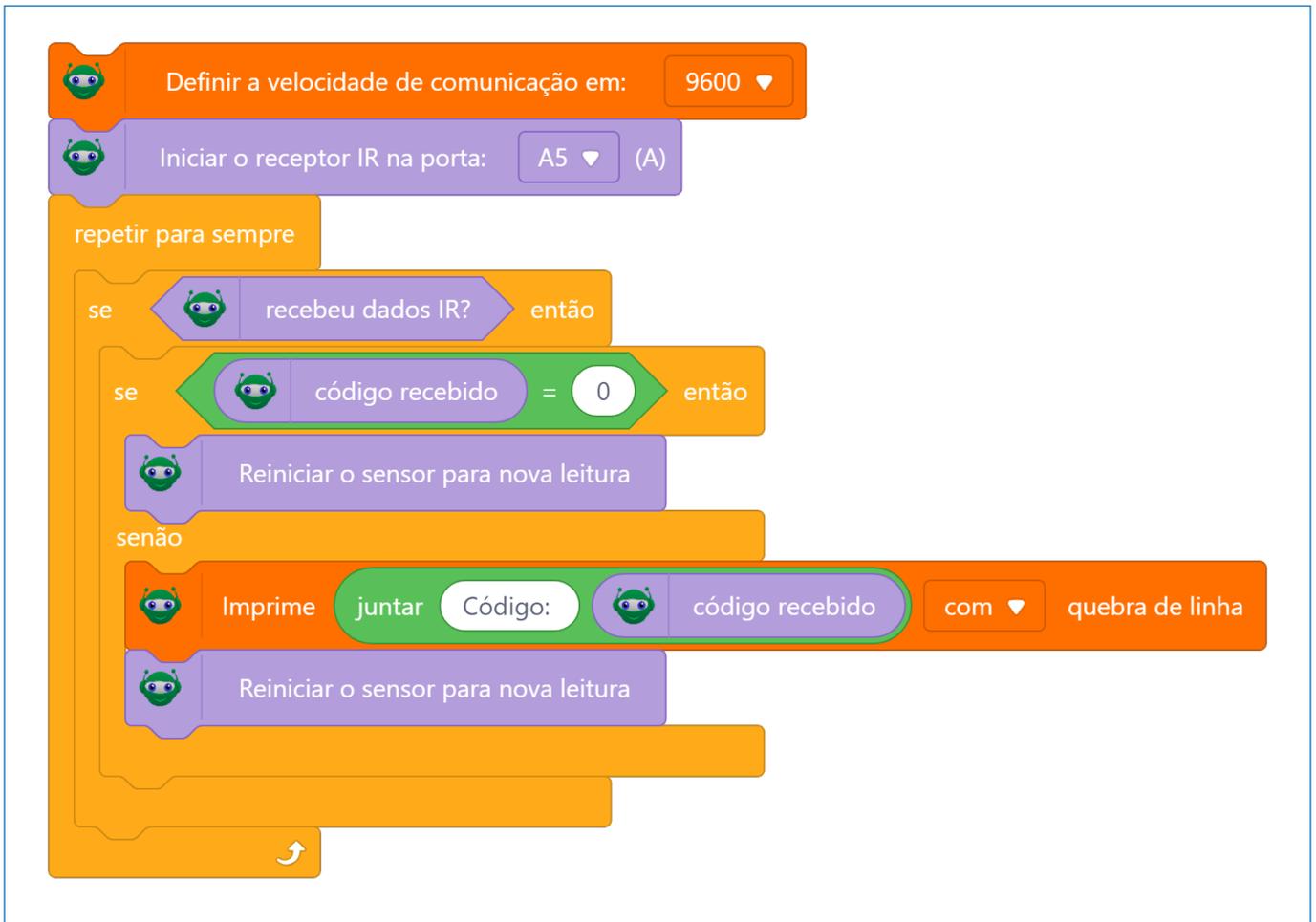
No mBlock, para definir a velocidade de comunicação, entre na aba "Monitor Serial" e arraste o bloco **<Definir a velocidade de comunicação em 9600>** depois, na aba "Receptor IR", arraste o bloco **<Iniciar o receptor IR na porta A5>**. Em seguida, arraste um laço **<repetir para sempre>** e na sequência a condicional para o recebimento dos dados do sensor IR **<se _____ então>**. Para programar o receptor IR você terá que baixar as extensões para esse componente, assim, na aba "extensão", digite o nome "Cleiton" em procurar, todas as extensões desse professor estarão disponíveis, em "Receptor IR", clique em adicionar. Após baixar, nessa aba, arraste o bloco **<recebeu dados IR?>** até o parâmetro da condicional **"se...então"**.

Para a segunda parte, arraste um bloco condicional **<se ___ então ___ se não>** e complete os parâmetros. Para o primeiro parâmetro, você precisará de um operador, na aba operadores, arraste **<___ = ___>**, no primeiro parâmetro, complete com o bloco **<código recebi-**

do> e no segundo parâmetro com o número "0". Novamente na aba "Receptor IR", arraste o bloco **<Reiniciar o sensor para nova leitura>** e para o "senão", na aba "Monitor serial", arraste o bloco **<Imprime _____ com quebra de linha>**. Na aba "operador", arraste o bloco **<juntar__ __>**, escreva "código" no primeiro parâmetro e no segundo arraste o bloco **<código recebido>** completando os parâmetros a serem impressos com quebra de linha. Para finalizar esta parte, arraste o bloco **<Reiniciar o sensor para a nova leitura>**.



Figura 12 - Exemplo de leitura de dados do sensor

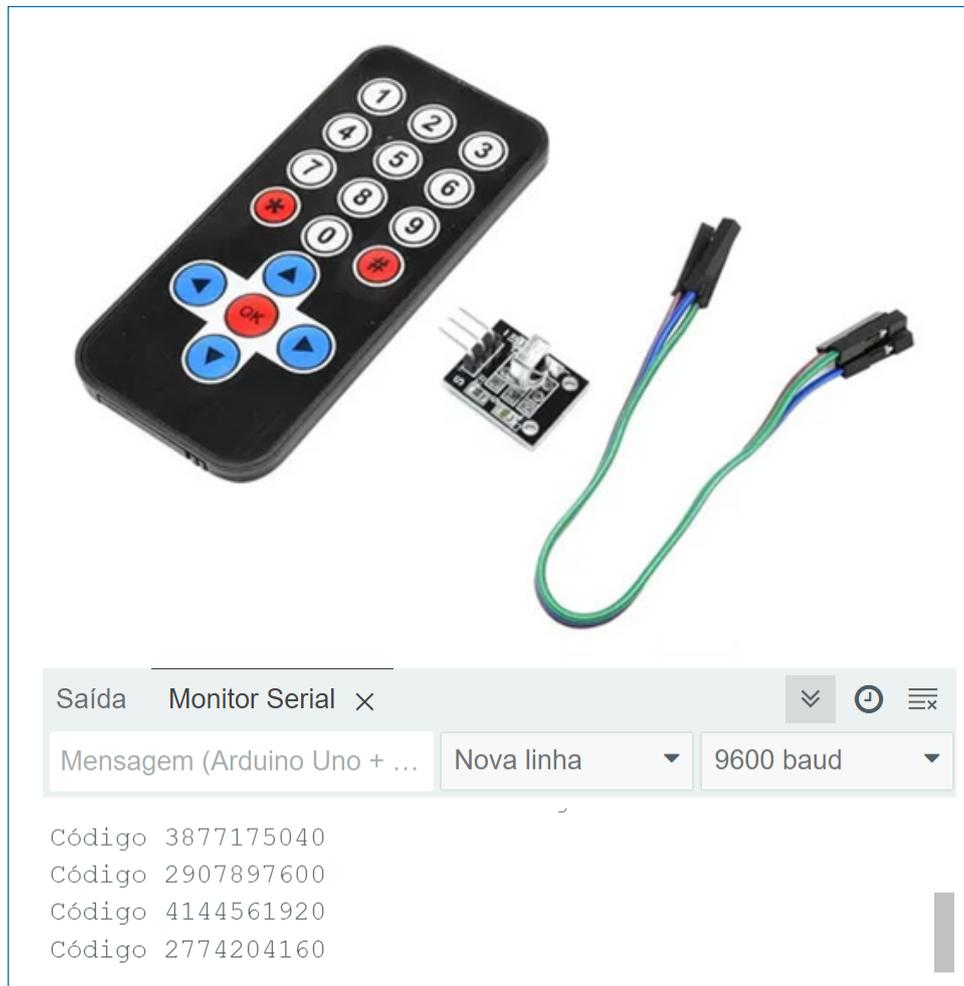


Fonte: mBlock, 2024.

Envie o código para o Arduino:

Após carregar o código no Arduino via mBlock, desconecte a placa do software para evitar conflitos. Em seguida, abra o Arduino IDE e acesse o Monitor Serial (geralmente localizado no canto superior direito). Sem a necessidade de abrir novos arquivos ou carregar mais códigos, você poderá visualizar os dados recebidos pelo Arduino, como os comandos enviados pelo controle infravermelho.

Figura 13 - Controle utilizado



No código acima, obtemos os códigos na seguinte ordem: seta para cima, seta para baixo, seta para esquerda e seta para a direita.

1	Seta para cima:	3877175040
2	Seta para baixo:	2907897600
3	Seta para a esquerda:	4144561920
4	Seta para a direita:	2774204160

Para a programação precisaremos somente desses botões. No código podemos definir a mesma numeração feita acima como convenção.

Esteira rolante - II

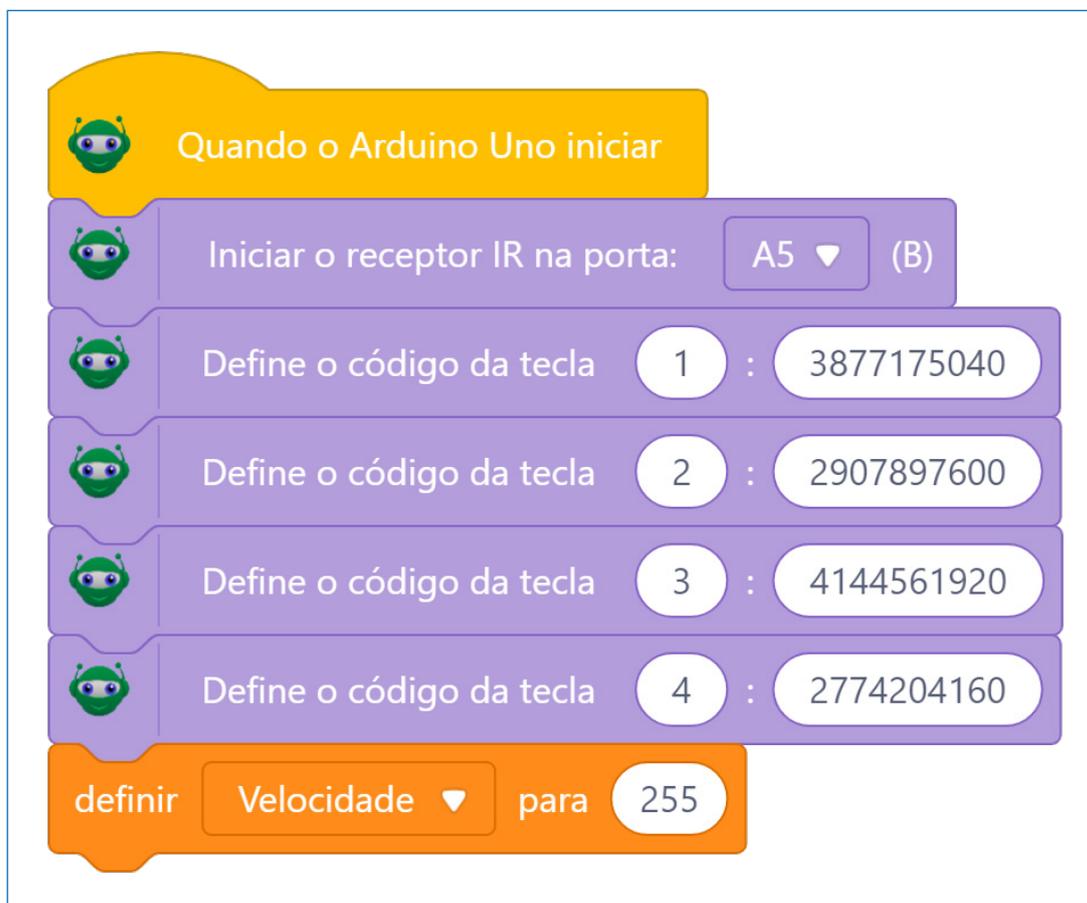
Nessa etapa, já conseguimos os códigos de cada botão que será utilizado. Agora, basta implementar o código em blocos referente ao funcionamento da esteira. Dividiremos em algumas partes para facilitar a compreensão.

A figura apresenta o início do novo código, onde declaramos as variáveis para controlar o sensor infravermelho (IR) na porta A5, configurando-o para a lógica B. Essa configuração é ideal para o funcionamento do motor DC. As teclas do controle remoto são representadas por números e seus respectivos códigos, obtidos na etapa anterior. Definimos a velocidade máxima do motor como 255.

Para isso, entre na aba “Eventos” e arraste o bloco de início **<Quando o Arduino Uno iniciar>**, os blocos para o receptor **<Iniciar o receptor IR na porta A5 (B)>** e definir os códigos para as teclas **<Define o código da tecla ___:___>**, preencha os parâmetros com os números da tecla e o código copiado do serial.

Na aba “Variáveis”, crie a variável “Velocidade”, arraste o bloco **<definir ___ para ___>** e preencha os parâmetros com “velocidade” e com a velocidade de “255”.

Figura 14 - Declaração das variáveis a ser utilizada no mBlock



Esteira rolante - II

Na sequência, implementaremos a função básica de ligar e desligar a esteira. Ao detectar um sinal do controle remoto, o código compara o código recebido com os códigos previamente definidos. Se o código corresponder ao botão “seta para cima”, a esteira é ligada na velocidade máxima. Caso contrário, se corresponder ao botão “seta para baixo”, a esteira é desligada.

Na aba “Controlo” arraste o bloco **<repetir para sempre>** e, na sequência, as condicionais **<se _____ então>** para o parâmetro, na aba “Receptor IR”, arraste o bloco **<recebeu dados IR?>**. Depois, mais uma vez, arraste duas condicionais **<se _____ então>** e complete os parâmetros. Na primeira condicional, na aba “Operadores” arraste o bloco **<_____ = _____>**, no primeiro parâmetro, arraste o bloco **<código recebido>** e no segundo **<código da tecla _____>** preencha com o número da tecla a ser apertado “1”, que corresponde ao código da seta para cima, e depois “2”, que corresponde ao código da seta para baixo. Na aba “portas” arraste o bloco **<definir a saída digital _____ em nível lógico _____>**, no primeiro parâmetro indica-se a porta, no segundo HIGH ou LOW, conforme Figura 15.

Figura 15 – Ligar e desligar a esteira



Esteira rolante - II

Nesse passo, adicionaremos a funcionalidade de acelerar e frear a esteira. Ao pressionar a seta para a esquerda no controle remoto, a velocidade do motor da esteira será reduzida em 10 unidades. Já ao pressionar a seta para a direita, a velocidade será aumentada em 10 unidades. Para evitar que a velocidade exceda o valor máximo de 255, incluímos uma verificação: se a velocidade calculada for maior que 255, ela é limitada a esse valor. Após cada verificação de código, o sensor infravermelho é reiniciado para detectar novos comandos do controle remoto.

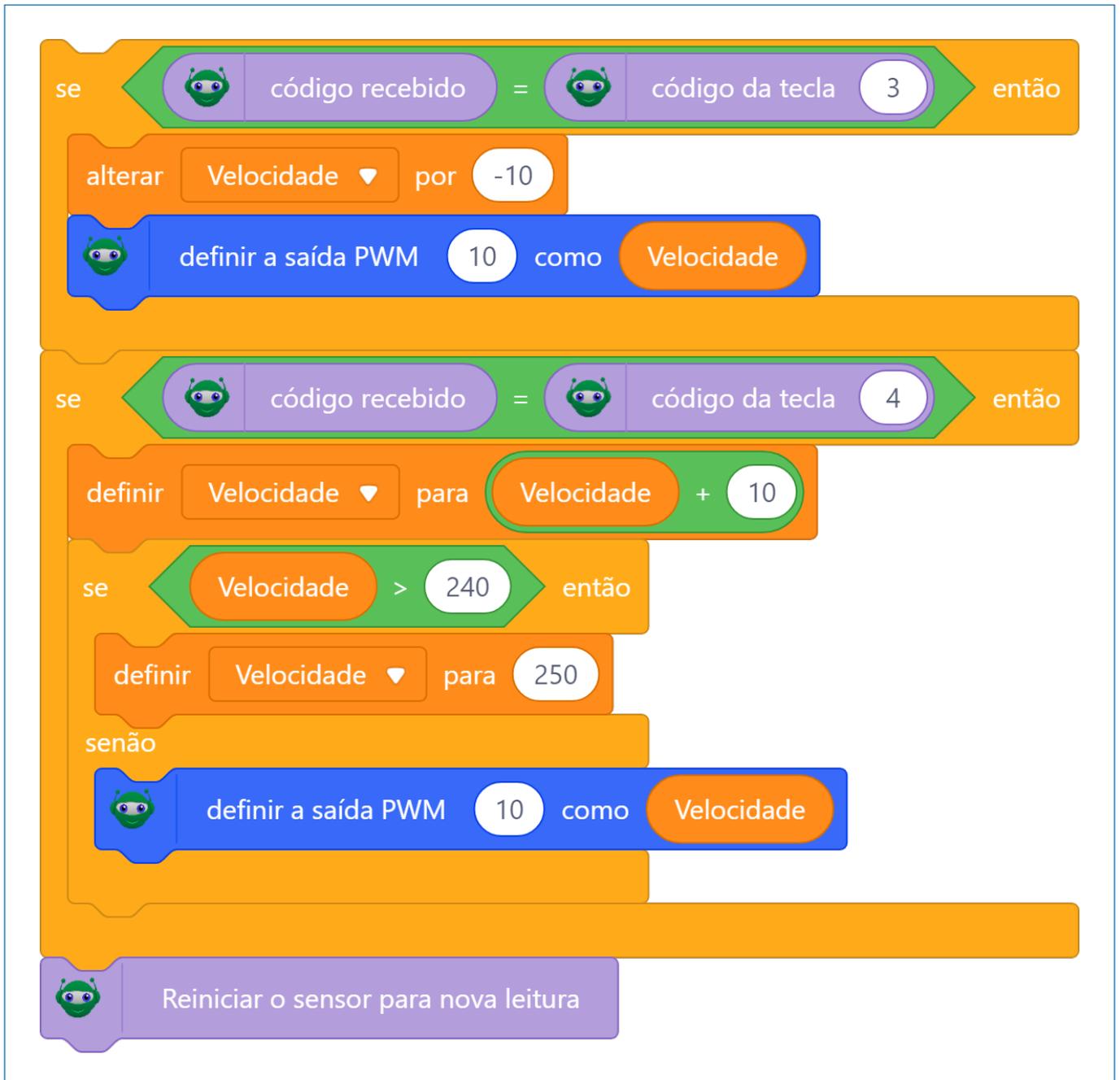
Novamente, na aba "Controlo" arraste o bloco **<se _____ então>** e em "Operadores" arraste o bloco **<_____ = _____>**. Para o primeiro parâmetro, arraste o bloco **<código recebido>** e no segundo **<código da tecla__>** indique o número 3, correspondente à tecla de seta da esquerda. Depois, em "Variável", arraste o bloco **<alterar Velocidade por __>** complete o parâmetro com o valor "10".

Para controlar a velocidade, entre na aba "Portas" e arraste o bloco **<definir a saída PWM__ como __>**, no primeiro parâmetro insira a velocidade "10" e no

segundo arraste a variável "Velocidade". Insira um segundo bloco **<se _____ então>** e em operadores arraste o bloco **<_____ = _____>**, no primeiro parâmetro insira o bloco do receptor IR **<código recebido>**, no segundo o bloco **<código da tecla__>**, indique a tecla "4", correspondente à tecla de seta da direita. Ainda dentro desta condicional, arraste o bloco de variável **<definir Velocidade para _____>**. Nesse parâmetro, arraste o operador **<_____ + _____>**, e preencha no primeiro a variável **<velocidade>** e no segundo insira o número da velocidade "10". Dentro dessa mesma condicional, insira outro bloco **<se _____ então _____ senão>**. Complete o primeiro parâmetro com o operador **<_____ > _____>** e inclua no operador a variável **<Velocidade>** e o valor de 240 no outro parâmetro. Ainda nessa condicional, arraste o bloco **<definir Velocidade para _____>**, inclua nesse parâmetro o número "255". Para o parâmetro do "senão", arraste novamente o bloco que indica a velocidade de saída PWM **<definir a saída PWM__ como __>**, no primeiro parâmetro insira a velocidade "10" e no segundo a variável **<Velocidade>**.



Figura 16 – Acelerar e frear a esteira



Na sequência, a programação completa.

Esteira rolante - II

Figura 17 - Programação completa

```
Quando o Arduino Uno iniciar
  Iniciar o receptor IR na porta: A5 (B)
  Define o código da tecla 1 : 3877175040
  Define o código da tecla 2 : 2907897600
  Define o código da tecla 3 : 4111561920
  Define o código da tecla 4 : 2774204160
  definir Velocidade para 255
  repetir para sempre
    se recebeu dados IR? então
      se código recebido = código da tecla 1 então
        definir a saída digital 10 em nível lógico HIGH
      se código recebido = código da tecla 2 então
        definir a saída digital 10 em nível lógico LOW
      se código recebido = código da tecla 3 então
        alterar Velocidade por -10
        definir a saída PWM 10 como Velocidade
      se código recebido = código da tecla 4 então
        definir Velocidade para Velocidade + 10
        se Velocidade > 240 então
          definir Velocidade para 255
        senão
          definir a saída PWM 10 como Velocidade
    Reiniciar o sensor para nova leitura
```

Desafios:

Que tal...

Inserir sensores ultrassônicos ou infravermelhos para detectar objetos próximos na esteira?

Inserir LEDs como indicadores de status para mostrar se a esteira está em funcionamento ou parada?

Inserir buzzer para a esteira emitir sons de alerta quando estiver com erros?

4. Feedback e finalização

Em relação ao projeto de esteira rolante desenvolvido:

1. A esteira funciona de forma consistente e confiável?
2. A esteira é capaz de realizar as tarefas para as quais foi projetada?
3. O controle da velocidade e direção da esteira é preciso?
4. O sistema de controle é robusto e resistente a falhas?
5. Quais são os componentes mais suscetíveis a falhas?
6. Quais melhorias poderiam ser feitas no projeto?

Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de Robótica.



REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27 mai. 2024.

WIKIPÉDIA. **Esteira transportadora**. https://pt.wikipedia.org/wiki/Esteira_transportadora. Acesso em: 07 ago. 2024.

WIKIPÉDIA. **Ponte H**. https://pt.wikipedia.org/wiki/Ponte_H. Acesso em: 07 ago. 2024.

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)
FACULDADE DE COMPUTAÇÃO (FACOM)

PROFESSORES

- Amaury Antônio de Castro Junior
- Anderson Corrêa de Lima
- Glauder Guimarães Ghinozzi
- Graziela Santos de Araújo
- Said Sadique Adi

ESTUDANTES (elaboração prévia)

- Filipe de Andrade Machado - Ciência da Computação
- Gabriel Alves Massuda Duarte - Engenharia de Computação
- José Augusto Lajo Vieira Vital - Ciência da Computação
- Lorena Valente Cavalheiro - Engenharia de Computação
- Matheus Kazumi Silva Miyashiro - Engenharia de Computação
- Nathalia dos Santos Melo - Engenharia de Software
- Yan Arruda Cunha - Engenharia de Computação
- Thiago Ferronato - Ciência da Computação
- Vitor Hugo dos Santos Duarte - Engenharia de Computação
- Wilker Sebastian Afonso Pereira - Ciência da Computação

ESTUDANTES (revisão)

- Arthur Henrique Andrade Farias - Ciência da Computação
- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Jenniffer Oliveira Checchia - Ciência da Computação
- Leonardo Vargas de Paula - Sistemas de Informação
- Marcos Gabriel da Silva Rocha - Engenharia de Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Nathanael Martins Wink - Ciência da Computação
- Victor Luiz Marques Saldanha Rodrigues - Ciência da Computação

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

