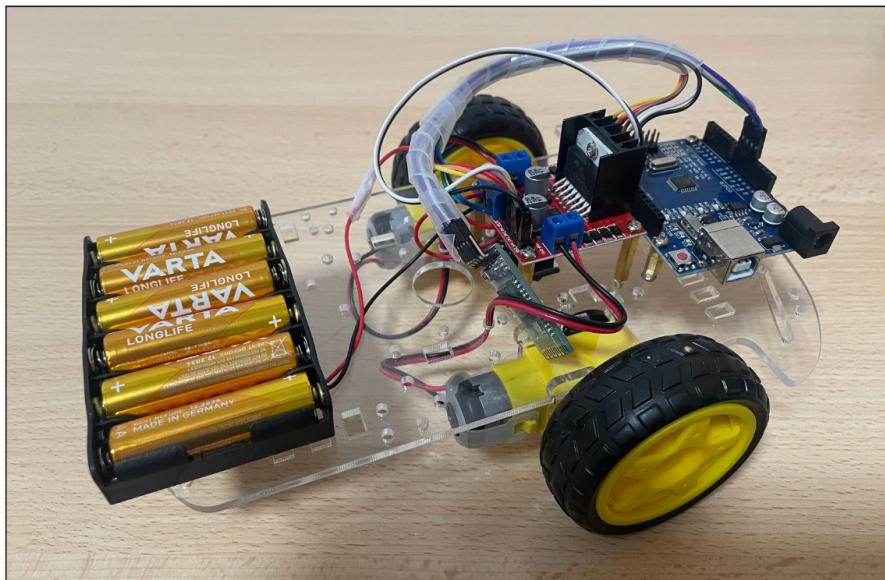


BRUMU



GUIA COMPLETO DE MONTAGEM E USO

Brumu: Kit educativo de baixo custo para apoiar o desenvolvimento do pensamento computacional em crianças

RESUMO

O projeto Brumu é uma proposta inovadora apresentada como defesa de Trabalho de Conclusão de Curso (TCC) pelos estudantes Bruna Teles e Munir Souza, sob orientação da professora Luciana Montera. Desenvolvido no âmbito da graduação em Sistemas de Informação da Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul, o projeto tem como objetivo incentivar o aprendizado de lógica de programação desde o ensino básico, utilizando um brinquedo educacional tecnológico que combina interatividade e criatividade para impactar e transformar o ensino de programação.

Este manual detalha o processo de montagem e operação de um carrinho robótico controlado por *Bluetooth*, utilizando a plataforma Arduino Uno como núcleo de controle. O carrinho é composto por motores controlados por uma ponte H, módulo *Bluetooth* para comunicação com dispositivos móveis e um chassis leve e resistente. A interação com o robô ocorre através de um aplicativo *mobile*, desenvolvido especificamente para este projeto, que permite o envio de comandos para controlar as direções e os movimentos do carrinho.

O documento inclui orientações completas para a construção do carrinho, desde a lista de componentes necessários e conexões eletrônicas até a programação do Arduino. Além disso, aborda a instalação e utilização do aplicativo *mobile*, facilitando a comunicação entre o usuário e o robô de maneira intuitiva. Com foco na democratização do acesso à robótica educacional, este manual visa permitir que usuários de diferentes níveis de habilidade técnica possam replicar e operar o projeto, promovendo o aprendizado prático de conceitos relacionados à robótica, automação e programação.

Palavras-chave: Resumo, Robótica Educacional, Arduino UNO, MIT App Inventor, Manual de Montagem e Uso.

SUMÁRIO

| | |
|---|-----------|
| 1 COMPONENTES PARA O CARRINHO..... | 4 |
| Antes de iniciar, certifique-se de possuir os seguintes itens:..... | 4 |
| TABELA 1 - Componentes para montagem do carrinho..... | 4 |
| 2 MONTAGEM DO CARRINHO..... | 6 |
| 2.1 ESTRUTURA FÍSICA..... | 6 |
| FIGURA 1 - Kit de chasse..... | 6 |
| FIGURA 2 - Motor fixado no chassi..... | 6 |
| FIGURA 3 - Roda instalada..... | 7 |
| FIGURA 4 - Roda boba instalada ao chassi..... | 7 |
| FIGURA 5 - Componentes fixados no chassi..... | 8 |
| FIGURA 6 - Suporte de pilhas fixado ao chassi..... | 8 |
| 2.2 CONEXÕES ELÉTRICAS..... | 9 |
| 3 CONFIGURAÇÃO E PROGRAMAÇÃO DO ARDUINO..... | 11 |
| 3.1 Instalação do Software Arduino IDE..... | 11 |
| 3.2 Conexão do Arduino ao Computador..... | 11 |
| 3.3 Código para Controle do Carrinho..... | 11 |
| 3.4 Carregamento do Código no Arduino..... | 12 |
| FIGURA 7 - Tela de edição de código do Arduino IDE..... | 12 |
| FIGURA 8 - Ícone para compilar código..... | 12 |
| FIGURA 10 - Ícone para realizar upload do código ao Arduino..... | 13 |
| 3.5 Testes Iniciais..... | 13 |
| 4 INSTALAÇÃO E USO DO APLICATIVO MOBILE..... | 14 |
| 4.1. Requisitos..... | 14 |
| 4.2. Download do Aplicativo..... | 14 |
| 4.3. Instalação do Aplicativo..... | 14 |
| 4.4. Uso do Aplicativo..... | 15 |
| FIGURA 12 - Tela do tabuleiro virtual..... | 15 |
| FIGURA 13 - Campos a serem alocados as peças de comandos..... | 16 |
| 4.5. Solução de Problemas..... | 16 |
| 4.6. Dicas de Uso..... | 16 |
| 5 PROGRAMAÇÃO DO APLICATIVO COM APP INVENTOR..... | 17 |
| 5.1. Requisitos..... | 17 |
| 5.2. Acessando o MIT App Inventor..... | 17 |
| 5.3. Importando o Código Fonte..... | 17 |
| 5.4. Explorando a Interface do App Inventor..... | 17 |
| 5.5. Modificando o Aplicativo..... | 18 |
| 5.6. Testando o Aplicativo..... | 18 |
| 5.7. Solução de Problemas..... | 18 |
| 5.8. Publicação e Compartilhamento..... | 19 |

1 COMPONENTES PARA O CARRINHO

Antes de iniciar, certifique-se de possuir os seguintes itens:

- **Componentes Eletrônicos:**

- 1x Placa Arduino Uno R3;
- 1x Módulo Bluetooth HC-05;
- 1x Ponte H L298N (controlador de motores);
- 2x Motores DC com engrenagem;
- 1x Suporte para 6 ou 4 pilhas AA;
- 6x Pilhas AA;
- 1x Botão Liga/Desliga;
- Fios jumper (macho-macho e macho-fêmea);

- **Componentes Mecânicos:**

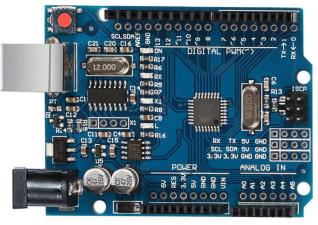
- 1x Chassi de acrílico para carrinho;
- 2x Rodas compatíveis com motores DC;
- 1x Roda boba (rodinha giratória);
- Parafusos e porcas (incluídos no kit do chassi);

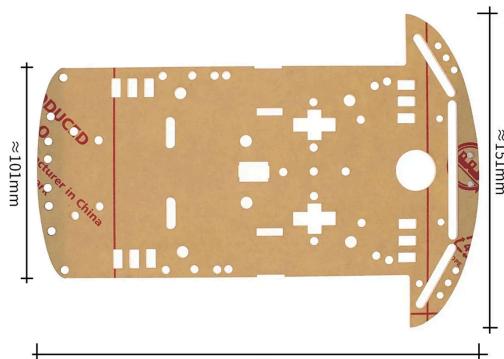
- **Ferramentas:**

- Chave de fenda;
- Ferro de solda e estanho (opcional, para maior estabilidade nas conexões);

Na Tabela 1, estão apresentadas imagens ilustrativas de cada componente necessário para a construção do carrinho Arduino. É importante destacar que, ao adquirir os materiais, especialmente no caso de um kit de chassi, você deve verificar cuidadosamente quais peças estão incluídas no kit e confirmar as especificações de cada item para garantir a compatibilidade com o projeto.

TABELA 1 - Componentes para montagem do carrinho

| | |
|--|---|
|  1x Placa Arduino Uno R3 |  1x Módulo Bluetooth HC-05 |
|--|---|

| | |
|--|--|
|  <p>1x Ponte H L298N (controlador de motores)</p> |  <p>2x Motores DC com engrenagem</p> |
|  <p>1x Suporte para 6 pilhas AA</p> |  <p>1x Botão Liga/Desliga</p> |
|  <p>Fios jumper (macho-macho e macho-fêmea)</p> |  <p>1x Chassi de acrílico ou madeira para carrinho</p> |
|  <p>2x Rodas compatíveis com motores DC</p> |  <p>1x Roda boba</p> |

2 MONTAGEM DO CARRINHO

2.1 ESTRUTURA FÍSICA

A Figura 1 ilustra as peças que geralmente acompanham um kit de chassis, bem como a forma correta de montar cada uma delas no conjunto, e como deverá ficar ao final da montagem.

FIGURA 1 - Kit de chasse



Aqui estão os passos detalhados para montar a estrutura física do carrinho robótico:

Passo 1: Fixação dos Motores.

1. Identifique os suportes laterais do chassis onde os motores serão montados.
2. Posicione os motores DC de forma que seus eixos fiquem alinhados com as aberturas destinadas para as rodas.
3. Utilize os parafusos e porcas fornecidos no kit para fixar firmemente cada motor ao chassis. Certifique-se de que estejam bem apertados para evitar movimentos indesejados durante o funcionamento.

FIGURA 2 - Motor fixado no chassi



Passo 2: Instalação das Rodas.

1. Pegue as rodas compatíveis com os motores e encaixe-as cuidadosamente nos eixos de cada motor.
2. Certifique-se de que as rodas estejam bem encaixadas para evitar que se soltem durante o uso.

FIGURA 3 - Roda instalada



Passo 3: Fixação da Roda Boba.

1. Localize o ponto traseiro do chassi destinado à instalação da roda boba (rodinha giratória).
2. Fixe a roda boba utilizando os parafusos fornecidos no kit do chassi. Essa peça é fundamental para garantir a estabilidade do carrinho, permitindo movimentos suaves em várias direções.

FIGURA 4 - Roda boba instalada ao chassi

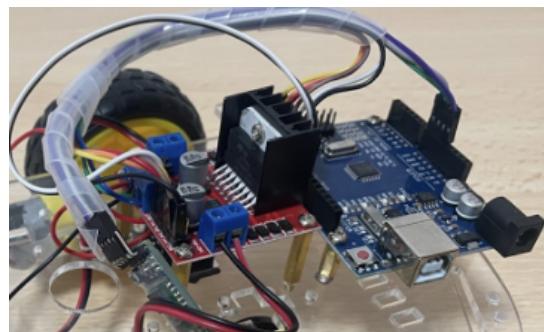


Passo 4: Posicionamento do Arduino e Componentes Eletrônicos.

1. Escolha uma área plana no chassi para fixar a placa Arduino Uno. Utilize fita dupla-face ou suportes próprios para fixá-la com segurança.

2. Repita o processo para fixar o módulo *Bluetooth* HC-05, o controlador de motores (ponte H L298N) e o suporte para as pilhas. Certifique-se de que todos os componentes estejam organizados para facilitar as conexões elétricas.

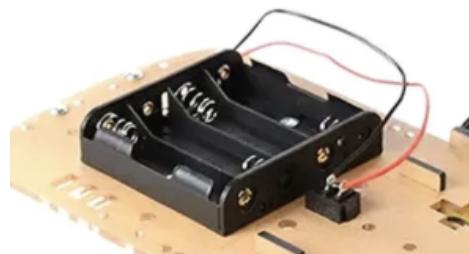
FIGURA 5 - Componentes fixados no chassi



Passo 5: Montagem da Fonte de Alimentação.

1. Instale o suporte para pilhas AA em uma posição central no chassi, garantindo equilíbrio e fácil acesso.
2. Conecte o botão Liga/Desliga ao suporte para pilhas, garantindo que ele possa ser usado para controlar a alimentação do carrinho.

FIGURA 6 - Suporte de pilhas fixado ao chassi



Passo 6: Organização Geral.

1. Verifique se todos os componentes estão firmemente fixados e alinhados corretamente.
2. Certifique-se de que o chassi esteja montado de forma equilibrada, com todas as peças instaladas corretamente para evitar instabilidade durante o uso.

Ao seguir cuidadosamente esses passos, você terá a estrutura física do carrinho pronta para as próximas etapas de conexão elétrica e programação.

2.2 CONEXÕES ELÉTRICAS

As conexões elétricas do carrinho são fundamentais para integrar os componentes e garantir o funcionamento adequado. Cada conexão deve ser feita com cuidado, seguindo as orientações a seguir:

Passo 1: Conexão dos Motores à Ponte H L298N.

1. Identificação das saídas da Ponte H (L298N):

- Localize os terminais de saída da ponte H:
 - **OUT1** e **OUT2** para o motor esquerdo.
 - **OUT3** e **OUT4** para o motor direito.

2. Ligação dos motores:

- Conecte os fios do motor esquerdo aos terminais **OUT1** e **OUT2**.
- Conecte os fios do motor direito aos terminais **OUT3** e **OUT4**.
- Verifique a polaridade.

* Se os motores girarem no sentido contrário, troque a ordem dos fios conectados.

Passo 2: Conexão dos Pinos de Controle da Ponte H ao Arduino.

1. Identificação dos pinos de controle:

- Os pinos **IN1**, **IN2**, **IN3** e **IN4** da ponte H controlam o sentido de rotação dos motores.

2. Ligação ao Arduino:

- Conecte:
 - **IN1** ao pino digital 2 do Arduino.
 - **IN2** ao pino digital 3 do Arduino.
 - **IN3** ao pino digital 4 do Arduino.
 - **IN4** ao pino digital 5 do Arduino.
 - **ENA** ao pino digital 9 do Arduino.
 - **ENB** ao pino digital 10 do Arduino.

3. Alimentação lógica da ponte H:

- Conecte o pino **5V** da ponte H ao pino **VIN** do Arduino.
 - Conecte o pino **GND** da ponte H ao pino **GND** do Arduino.
-

Passo 3: Conexão do Módulo *Bluetooth* HC-05.

1. Alimentação:

- Conecte o pino **VCC** do módulo *Bluetooth* ao pino **5V** do Arduino.
- Conecte o pino **GND** do módulo ao **GND** do Arduino.

2. Comunicação Serial:

- Conecte:

-
- O pino **RX** do módulo *Bluetooth* ao pino digital 7 do Arduino.
 - O pino **TX** do módulo *Bluetooth* ao pino digital 6 do Arduino.
-

Passo 4: Alimentação do Sistema.

1. Conexão da Fonte de Energia:

- Conecte o terminal negativo do suporte de pilhas (fio preto) ao terminal **GND** da ponte H e também ao pino **GND** do Arduino.

2. Botão Liga/Desliga:

- Conecte o terminal positivo do suporte de pilhas (fio vermelho) ao terminal do conector negativo, do botão de ligar e desligar.
 - Conecte com um fio o botão do lado positivo, ao terminal **+12V** da ponte H. Isso permitirá ligar e desligar o sistema facilmente.
-

Passo 5: Revisão Geral.

1. Organização dos fios:

- Utilize abraçadeiras ou fita isolante para organizar os fios, evitando que se soltem ou se enrolem nas rodas.

2. Teste das conexões:

- Verifique com cuidado cada conexão, garantindo que estejam firmes e nos terminais corretos.
 - Certifique-se de que não há fios desencapados em contato que possam causar curto-circuitos.
-

Ao finalizar essas conexões, seu carrinho estará eletricamente preparado para ser programado e controlado pelo aplicativo. Certifique-se de realizar testes básicos antes de iniciar o uso completo para garantir que tudo está funcionando conforme esperado.

3 CONFIGURAÇÃO E PROGRAMAÇÃO DO ARDUINO

Nesta seção, abordaremos os passos necessários para configurar e programar o Arduino Uno, garantindo o funcionamento correto do carrinho. A programação define o comportamento dos motores e a comunicação *Bluetooth* para o controle via aplicativo *mobile*.

3.1 Instalação do Software Arduino IDE

1. Download do Arduino IDE:

- Acesse o site oficial do Arduino (<https://www.arduino.cc/>) e baixe a versão mais recente do software para o seu sistema operacional (Windows, macOS ou Linux).

2. Instalação:

- Siga as instruções fornecidas no site para instalar o Arduino IDE em seu computador.

3. Drivers USB:

- Certifique-se de que os drivers USB do Arduino estão instalados corretamente. Isso geralmente ocorre automaticamente durante a instalação da IDE.

3.2 Conexão do Arduino ao Computador

1. Cabo USB:

- Conecte o Arduino Uno ao computador utilizando um cabo USB (tipo A para tipo B).

2. Seleção da Porta:

- Abra o Arduino IDE, vá até o menu **Ferramentas > Porta** e selecione a porta correspondente ao Arduino (geralmente identificada como "COMx" no Windows ou "/dev/ttyUSBx" no Linux/macOS).

3. Seleção da Placa:

- No menu **Ferramentas > Placa**, selecione "Arduino Uno" para configurar a IDE para o modelo correto.

3.3 Código para Controle do Carrinho

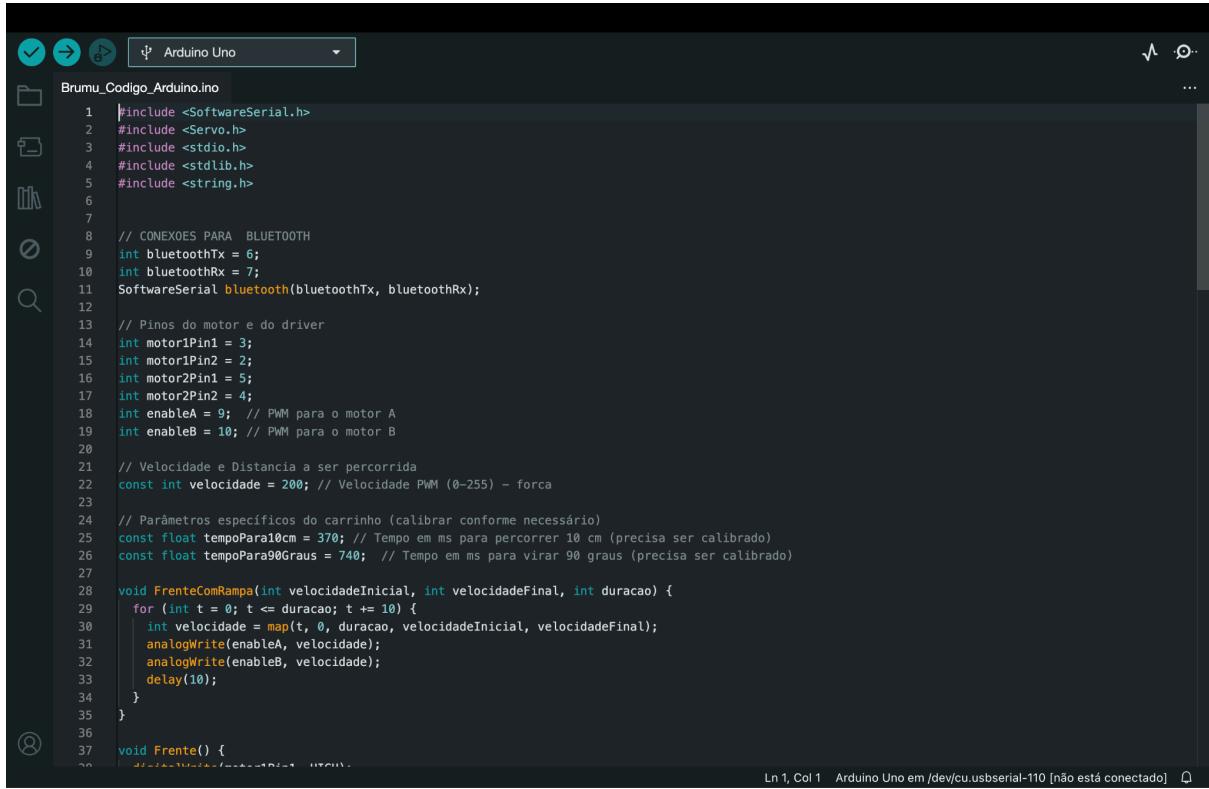
O código que deverá ser carregado no Arduino para controlar os motores do carrinho e processar os comandos recebidos pelo módulo *Bluetooth* está disponível publicamente no seguinte repositório do GitHub: <https://github.com/trunabeles/Brumu> , dentro da pasta nomeada “Arquivos do Arduino”.

3.4 Carregamento do Código no Arduino

1. Abra o Código na IDE:

- Copie e cole o código presente no repositório do GitHub no editor de texto do Arduino IDE, conforme Figura 7.

FIGURA 7 - Tela de edição de código do Arduino IDE



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Shows "Arduino Uno" and a dropdown menu.
- File Menu:** Includes options like File, Edit, Tools, Sketch, Help, and a Recent Projects section.
- Code Area:** Displays the code for "Brumu_Codigo_Arduino.ino". The code includes headers for SoftwareSerial, Servo, stdio, stdlib, and string. It defines pins for Bluetooth (bluetoothTx=6, bluetoothRx=7), motors (motorPin1=3, motorPin2=2, motor2Pin1=5, motor2Pin2=4), and PWM pins (enableA=9, enableB=10). It also defines speeds (velocidade = 200) and timing constants (tempoPara10cm = 370ms, tempoPara90Graus = 740ms). A function "FrenteComRampa" uses a map function to interpolate between initial and final speeds over a duration. A "Frente()" function calls "FrenteComRampa" with specific parameters.
- Status Bar:** Shows "Ln 1, Col 1" and "Arduino Uno em /dev/cu.usbserial-110 [não está conectado]".

2. Verifique o Código:

- Clique no botão de verificação (Figura 8) na parte superior da IDE para verificar se há erros de sintaxe. Após isso, será apresentado a compilação do mesmo, conforme Figura 9.

FIGURA 8 - Ícone para compilar código



FIGURA 9 - Compilação do código para verificação

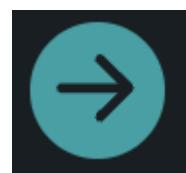
```
21 // Velocidade e Distancia a ser percorrida
22 const int velocidade = 200; // Velocidade PWM (0-255) - forca
23
Salida
El Sketch usa 5860 bytes (18%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 371 bytes (18%) de la memoria dinámica, dejando 1677 bytes para las variables locales. El máximo es 2048 bytes.

Compilação concluída.
```

3. Envie o Código:

- Clique no botão de upload (ícone de seta para a direita - Figura 10) para enviar o código para o Arduino. Aguarde a conclusão do carregamento.

FIGURA 10 - Ícone para realizar upload do código ao Arduino



4. Confirmação:

- Após o upload, o Arduino começará a executar automaticamente o código enviado.

3.5 Testes Iniciais

1. Alimente o Sistema:

- Conecte o suporte de pilhas ao Arduino e ligue o botão Liga/Desliga.

2. Teste os Motores:

- Verifique se os motores respondem corretamente aos comandos enviados pelo módulo *Bluetooth*.

3. Depuração:

- Se algo não funcionar como esperado, verifique as conexões elétricas e os pinos definidos no código.

Com essa configuração e programação, o Arduino estará pronto para controlar os motores do carrinho e receber comandos do aplicativo *mobile*, permitindo o controle remoto do robô.

4 INSTALAÇÃO E USO DO APLICATIVO MOBILE

O aplicativo *mobile* desenvolvido é uma parte essencial para o controle do carrinho robótico. Ele permite que o usuário envie comandos ao robô através de uma interface *Bluetooth* simples e intuitiva. Siga as etapas abaixo para instalar e utilizar o aplicativo.

4.1. Requisitos

Antes de começar, certifique-se de que você possui:

- Um dispositivo Android com *Bluetooth* habilitado.
 - O arquivo **.apk** do aplicativo.
 - O carrinho robótico montado e programado, com o módulo *Bluetooth* HC-05 configurado.
-

4.2. Download do Aplicativo

1. Obtenção do Arquivo **.apk**:

- Acesse o repositório GitHub do projeto Brumu (<https://github.com/trunabeles/Brumu>) e faça o download do arquivo **.apk** para o seu dispositivo Android.
 - Arquivo presente dentro da pasta chamada “Arquivos do Aplicativo”.
-

4.3. Instalação do Aplicativo

1. Permitir Fontes Desconhecidas:

- No dispositivo Android, vá para **Configurações > Segurança > Fontes Desconhecidas** e ative a opção para permitir a instalação de aplicativos que não são da Google Play Store.

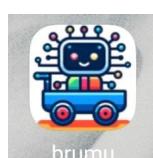
2. Instalar o Aplicativo:

- Localize o arquivo **.apk** no dispositivo (geralmente na pasta "Downloads").
- Toque no arquivo e siga as instruções para concluir a instalação.

3. Confirmação de Instalação:

- Após a instalação, você verá o ícone do aplicativo (Figura 11) na tela inicial do dispositivo, a exemplo abaixo:

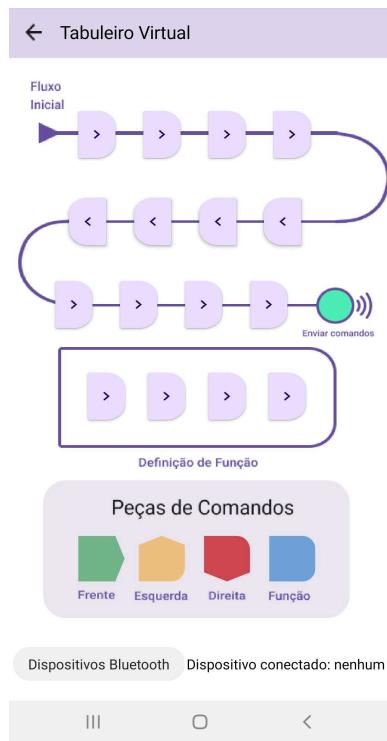
FIGURA 11 - Ícone do Aplicativo



4.4. Uso do Aplicativo

1. **Configuração Inicial:**
 - Ligue o carrinho e certifique-se de que o módulo *Bluetooth* está ativo.
 - Abra o aplicativo no dispositivo Android.
2. **Emparelhamento Bluetooth:**
 - No dispositivo Android, vá para **Configurações > Bluetooth** e procure pelo dispositivo Bluetooth "HC-05".
 - Selecione o dispositivo e insira o código de emparelhamento (geralmente "1234" ou "0000").
3. **Conexão com o Carrinho:**
 - No aplicativo, na tela do Tabuleiro Virtual (Figura 12), toque no botão "Dispositivos Bluetooth" e selecione o dispositivo "HC-05" na lista de dispositivos disponíveis.

FIGURA 12 - Tela do tabuleiro virtual

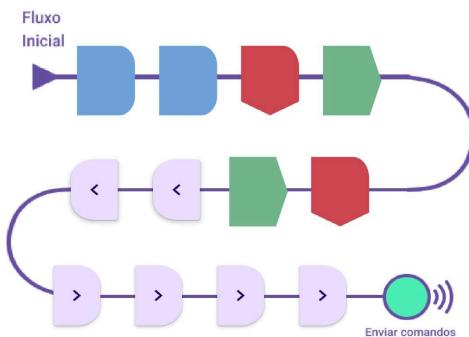


4. Controle do Carrinho:

- Use os botões disponíveis no aplicativo para controlar os movimentos do carrinho:
 - **Frente:** Avança o carrinho.
 - **Esquerda:** Gira o carrinho para a esquerda.
 - **Direita:** Gira o carrinho para a direita.
 - **Função:** Chama a função definida.

- Para isso, as peças de comandos trabalham com a ação de arrasta e solta, sendo assim, conforme a Figura 13, basta arrastar cada uma ao campo que desejem ao fluxo de comandos, seguindo a linha de execução apresentada em tela.

FIGURA 13 - Campos a serem alocados as peças de comandos



5. Definição de Funções (se aplicável):

- Utilize a funcionalidade de "Definição de Funções" para criar comandos compostos que podem ser executados de forma integrada, otimizando trajetos mais longos.
-

4.5. Solução de Problemas

1. O aplicativo não conecta ao Bluetooth:

- Verifique se o dispositivo "HC-05" está ligado e emparelhado corretamente.
- Certifique-se de que o módulo *Bluetooth* do carrinho está funcionando.

2. Os comandos não funcionam:

- Confirme se os pinos RX/TX do módulo *Bluetooth* estão conectados corretamente ao Arduino.
- Verifique se o código do Arduino está carregado e funcionando.

3. O carrinho não responde:

- Certifique-se de que a alimentação elétrica do carrinho está adequada e os motores estão conectados corretamente.
-

4.6. Dicas de Uso

- Mantenha o dispositivo Android a uma distância máxima de 10 metros do carrinho para garantir a conexão estável do *Bluetooth*.
- Experimente criar diferentes trajetos e testar as funcionalidades avançadas, como loops de comandos ou sequências otimizadas.

5 PROGRAMAÇÃO DO APLICATIVO COM APP INVENTOR

Caso deseje personalizar ou aprofundar a programação do aplicativo *mobile* do kit Brumu, o MIT App Inventor é uma excelente plataforma para isso. Ele permite modificar o código fonte disponibilizado no GitHub e criar novas funcionalidades ou interfaces.

Logo abaixo, serão listados alguns passos iniciais que poderão lhe apoiar nesse processo, desde a importação do código até a realização de personalizações e testes no aplicativo.

5.1. Requisitos

Antes de começar, certifique-se de ter:

- Um computador ou dispositivo com acesso à internet.
- Uma conta gratuita no MIT App Inventor (<https://appinventor.mit.edu/>).
- O arquivo do código fonte do aplicativo (.aia) disponível no repositório do projeto no GitHub (<https://github.com/trunabeles/Brumu/tree/main>).

5.2. Acessando o MIT App Inventor

1. Acesse o site oficial do MIT App Inventor (<https://appinventor.mit.edu/>).
2. Clique em **Create Apps** e faça login com sua conta Google.

5.3. Importando o Código Fonte

1. Download do Código Fonte:

- Acesse o repositório do projeto Brumu no GitHub (<https://github.com/trunabeles/Brumu/tree/main>).
- Baixe o arquivo do projeto do aplicativo (.aia).

2. Importar no App Inventor:

- No App Inventor, clique em Projects > Import project (.aia) from my computer.
- Selecione o arquivo .aia baixado e clique em OK para carregá-lo.

5.4. Explorando a Interface do App Inventor

O App Inventor é dividido em duas principais áreas de trabalho:

1. Designer (Interface Gráfica):

- Aqui você pode modificar o layout do aplicativo, adicionando ou removendo botões, imagens e outros elementos visuais.

- Exemplo: Personalizar a tela inicial, alterar o texto dos botões ou adicionar novos elementos, como sliders para controle de velocidade.

2. Blocks (Lógica de Programação):

- Este é o ambiente de programação visual, onde você cria a lógica do aplicativo usando blocos.
- Exemplo: Adicionar um novo botão que envia comandos específicos ao robô ou modificar a sequência de envio de comandos *Bluetooth*.

5.5. Modificando o Aplicativo

1. Alteração na Interface Gráfica:

- Vá para a aba Designer.
- Clique em um elemento da interface para alterar suas propriedades (ex.: texto, cor, tamanho).
- Adicione novos elementos a partir da barra lateral (ex.: novos botões ou imagens).

2. Adição de Novas Funcionalidades:

- Vá para a aba Blocks.
- Arraste e conecte blocos para definir novas funcionalidades.
- Exemplo: Criar um botão que envia um comando para o robô executar uma rotação específica.

5.6. Testando o Aplicativo

1. Teste em Tempo Real:

- Baixe o aplicativo de suporte MIT AI2 Companion no seu dispositivo Android.
- No App Inventor, clique em Connect > AI Companion e escaneie o QR Code exibido.
- As alterações no aplicativo serão refletidas em tempo real no dispositivo.

2. Exportando e Testando:

- Após finalizar as alterações, clique em Build > App (save .apk to my computer) para gerar o arquivo .apk.
- Instale o aplicativo no dispositivo Android e teste as novas funcionalidades.

5.7. Solução de Problemas

• O projeto não carrega no App Inventor:

- Verifique se o arquivo .aia foi baixado corretamente.
- Confirme se o App Inventor está atualizado e funcionando.

• O aplicativo não funciona como esperado:

- Revise as conexões dos blocos e certifique-se de que estão lógicos.
- Teste cada funcionalidade individualmente antes de gerar o .apk.

5.8. Publicação e Compartilhamento

1. Após concluir suas personalizações, você pode compartilhar o projeto modificado:
 - Exporte o arquivo .aia atualizado em Projects > Export selected project (.aia).
 - Compartilhe o arquivo para que outros possam importar e usar suas melhorias.
2. Publique o aplicativo em lojas de aplicativos, como a Google Play, para facilitar o acesso de outros usuários.