



Associação Propagadora Esdeva
Centro Universitário Academia - UniAcademia
Curso de Engenharia Elétrica
Artigo do Projeto de Iniciação Científica

TÍTULO: ROBÔ CONTROLADO REMOTAMENTE VIA ARDUINO

*Alunos: Andrew Furtado Salles Toledo; Laura Grossi de Oliveira Souza;
Renan Neves de Souza Assis
Prof. Orientador: Victor Fernandes
Centro Universitário Academia, Juiz de Fora, MG*

Linha de Pesquisa: Automação

RESUMO

Esse trabalho visa desenvolver um robô controlado remotamente a partir do Arduino e mostrar as etapas do desenvolvimento de um protótipo bem como sua viabilidade de custos. Durante o projeto, foram utilizadas ferramentas de simulação para auxiliar na escolha de componentes e para otimizar o *script* a ser implementado no Arduino. Por fim, foi possível analisar qual loja oferta o melhor custo-benefício para a prototipagem do robô.

Palavras-chave: Arduino. Tinkercad. Tutorial. Custo.

ABSTRACT

This work aims to develop a remotely-controlled robot by Arduino and to show the stages of the development of a prototype as well as its cost viability. During the project, simulation tools were used to assist in the choice of components and to optimize the script to be implemented in Arduino. Finally, it was possible to analyze which store offers the best cost-benefit ratio for robot prototyping.

Keywords: Arduino. Tinkercad. Tutorial. Associated Cost.

1 INTRODUÇÃO

Embora pareçam simples, muitos equipamentos residenciais tais como televisão, micro-ondas, máquinas de lavar, entre outros, possuem circuitos complexos que envolvem sistemas lógicos responsáveis por controlar a execução da sua função.

Com isso é possível observar uma dependência e aumento significativo da tecnologia e conseqüentemente acréscimo no uso de plataformas de prototipagem para microcontroladores, permitindo assim a prática de desenvolvimento de soluções próprias baseadas em sistemas eletrônicos. Esse é o caso da plataforma que será abordada ao longo deste artigo, o Arduino.

O Arduino é uma placa de circuito microcontrolador muito versátil, permitindo gravação de instruções, em um *script*, por uma linguagem de programação de baixa complexidade, podendo interagir com o ambiente por meio de *hardware* e *software*. Conta com uma interface programável pelo computador, permitindo a configuração de periféricos que podem realizar determinadas ações, desde o movimento de um braço mecânico até o controle e a comunicação de um sensor sem fio. Um exemplo simples, para um maior entendimento, seria acender uma luz por um intervalo de tempo depois que um botão é pressionado. Nesse exemplo, a lâmpada e o botão seriam os periféricos conectados ao Arduino.

Já na programação do Arduino é utilizado a IDE, um *software* no qual é possível escrever o código na linguagem de programação que o Arduino interpreta, o *C-like*, uma linguagem baseada em C. Sendo um conjunto de instruções, das quais é feito o upload para o Arduino, e assim, ele será capaz de executar as instruções que foram programadas, interagindo com o que estiver conectado a ele. Também, o Arduino pode ser conectado a placas eletrônicas, como receptores GPS, displays de LCD, etc., que podem interagir e expandir suas diversas funções, essas placas são conhecidas como Shields (escudos). Vale ressaltar que não é necessária a utilização do Shield, pois é possível executar a mesma função através de uma matriz de contatos (*protoboard*) ou da placa de circuito impresso (PCB). Para começar a utilizar o Arduino é preciso acessar o site (ARDUINO, 2020) e baixar o *software* para Mac, PC ou Linux. Há diversas placas de Arduino que são destinadas a muitos tipos de aplicação, sendo que todas podem ser programadas utilizando o mesmo *software* de desenvolvimento e, em geral, os programas que funcionam em uma placa também funcionam em todas as demais.

Durante o desenvolvimento do projeto foi utilizada a plataforma Tinkercad (TINKERCAD, 2020), uma ferramenta online e gratuita de simulação de circuitos elétricos. Com ela é possível projetar peças das mais variadas geometrias de uma forma intuitiva e acessível, apenas criando uma conta no site gratuitamente. Além da possibilidade de usar a ferramenta online, é possível fazer o download do Tinkercad

até mesmo no iPad, basta conectar-se à internet para usá-lo. Esse *software* conta com recursos bem básicos, mas que são ótimos para quem está iniciando a modelagem 3D. Usar uma ferramenta simples e gratuita como o Tinkercad é uma ótima forma de começar na modelagem 3D.

Através do Arduino, um projeto em que o LED vermelho é utilizado para emitir uma sequência de S.O.S. repetidamente foi realizado (MONK, 2014). Os autores fizeram também vários outros projetos, dentre os quais destacamos o osciloscópio, onde valores analógicos de entradas foram lidos e repassados ao computador através do cabo USB. Além disso, uma aplicação *web* foi utilizada para realizar o controle sobre a abertura e fechamento do portão automático, controle da iluminação residencial, entre outros (CAMPOS, 2014). Essa aplicação é capaz de enviar dados a uma interface de *hardware* através de uma comunicação entre o computador e o Arduino. Por fim, um protótipo baseado em Arduino foi utilizado para o auxílio a deficientes visuais no transporte público, em que o deficiente visual foi capaz de solicitar um ônibus utilizando um teclado numérico em braile com sinais sonoros para acessibilidade (COLARES & LIMA, 2018).

Diferentemente dos trabalhos anteriores, neste artigo abordaremos a placa Arduino, apresentando alguns de seus componentes e como você pode utilizá-lo nos projetos. Explicaremos também sobre a plataforma Tinkercad, uma ferramenta de simulação, que foi utilizada em nosso trabalho para a prototipagem do robô e controle. Além disso, indicaremos onde conseguir os materiais para construção dos protótipos e discutiremos sobre a loja que terá um preço mais acessível para a compra.

A Seção 2 aborda o desenvolvimento deste artigo, onde é introduzido o *software* Tinkercad e também são apresentados os principais componentes do projeto bem como o funcionamento dos mesmos. A Seção 3 detalha os resultados obtidos e a discussão acerca deles. Por fim, a conclusão é apresentada na Seção 4.

2 DESENVOLVIMENTO

O Tinkercad é um *software* de modelagem 3D, desenvolvido pela Autodesk. Por ser um programa gratuito e que pode ser acessado diretamente através do navegador de Internet, é uma ótima ferramenta de aprendizagem para alunos e professores. Esse *software* faz com que a modelagem virtual do circuito seja similar a

forma com a qual o circuito seria montado na prática, sendo possível utilizar de uma gama de equipamentos de um típico laboratório de eletrônica, tais como *protoboards*, multímetros e fontes de energia. Além disso, é possível simular o funcionamento do Arduino, permitindo realizar suas conexões com os demais eletrônicos na *protoboard* e, também, escrever um conjunto de instruções e comandos a serem executadas (esse conjunto é conhecido como *script*).

Durante o desenvolvimento do projeto foi utilizado o Tinkercad para simular o funcionamento dos componentes do controle e do robô, testar tipos de conexões entre o Arduino e o controle remoto e validar o *script* utilizado. Com isso, conseguimos analisar quais componentes seriam mais compatíveis, definir o tipo de comunicação entre o controle remoto e o robô e otimizar o *script* do Arduino.

A seguir, serão apresentados os principais componentes utilizados durante o projeto:

a) Arduino Uno

O Arduino Uno (Figura 1) é uma placa microcontroladora baseada no microcontrolador da Atmel, ATmega328P. Em italiano "Uno" quer dizer um, e foi utilizado para o lançamento da primeira versão "Arduino 1.0". A placa possui um hardware de código aberto, ou seja, você pode construir sua própria placa. Possui 14 pinos de entradas e saídas digitais e mais 6 entradas e saídas do tipo analógica, prontas para uso (não necessita de configuração prévia). A mesma já possui todos os componentes necessários para o seu funcionamento, como conversores, resistores, diodos e dispositivos de proteção, ou seja, não é preciso montar uma placa específica como eram feitos utilizando apenas os microprocessadores. Para mais informações a respeito do Arduino Uno, consultar a Tabela 1.

Para sua utilização e programação é necessário apenas um cabo USB e um computador com a plataforma IDE do Arduino instalada. Lembrando que a IDE é gratuita e pode ser encontrada no site do próprio fabricante (SOFTWARE, 2020).



Figura 1 - Arduino Uno SMD CH340.

Tabela 1 - Especificações Técnicas do Arduino Uno.

Microcontrolador	ATmega328P
Tensão operacional	5 V
Tensão de entrada	7-12 V
Tensão de entrada (limite)	6-20 V
Pinos de I/O digitais	14 (6 PWM)
Pinos de I/O digital PWM	6
Pinos de entrada analógica	6
Corrente CC por pino de I/O	20 mA
Corrente CC para pino de 3,3 V	50 mA
Memória flash	32 kB (ATmega328P)
SRAM	2 kB (ATmega328P)
EEPROM	1 kB (ATmega328P)
Velocidade do clock	16 MHz
LEDs internos	13
Comprimento x Largura	68,6 x 53,4 mm

b) Módulo Wireless NRF24L01

O módulo NRF24L01 (Figura 2) tem custo reduzido e é capaz de enviar e receber sinal através de uma rede sem fio. Produzido pela *Nordic Semiconductor*, o mesmo trabalha na faixa de frequência de 2,4 GHz e com uma taxa de dados de até 2 Mbps (NORDIC, 2020), podendo chegar a uma distância de 100 m em espaço aberto. O formato da ligação de suas conexões na placa o impossibilitam de ligá-lo diretamente na *protoboard*, sendo aconselhável o uso de adaptador ou *jumpers* do tipo macho-fêmea.

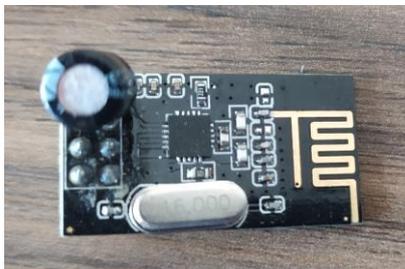


Figura 2 - Módulo Wireless NRF24L01 de 2,4GHz.

c) PWM

A modulação por largura de pulso, ou PWM, é uma técnica que permite o controle da energia média de consumo através da geração de pulsos quadrados com intervalo de tempo variável (HIRZEL, 2018). Em outras palavras, através do controle do tempo em que uma onda quadrada fica em nível lógico alto, podemos alterar sua amplitude média, mantendo uma mesma frequência. O tempo em que o sinal está em nível lógico alto é chamado de ciclo ativo ou *duty cycle* e pode ser calculado como:

$$Duty\ Cycle\ (\%) = \frac{T_{on}}{T_{on} + T_{off}} \times 100,$$

onde, T_{on} e T_{off} são o tempo em condução e o tempo de corte, respectivamente, e ambos são dados em segundos.

No Arduino, conseguimos encontrar portas com PWM, variando a quantidade de acordo com o modelo do Arduino. Na Figura 3, podemos ver as portas PWM de um Arduino Uno. Observe que de as saídas PWM são as marcadas com um “~”.



Figura 3 - Portas PWM do Arduino Uno.

As saídas PWM podem ser controladas através de códigos. Para sua programação devemos utilizar uma função específica, o `analogWrite()`, (ANALOGWRITE, 2020). Essa função possui a seguinte sintaxe: `analogWrite(pino, valor)`; o pino é a porta escolhida no Arduino e o valor corresponde ao *duty cycle*, normalizado por 255. Por exemplo, para o pino 3 utilizar um *duty cycle* de 127/255, devemos usar “`analogWrite(3, 127)`”.

d) Ponte H

Ponte H é um circuito eletrônico de potência, que tem por base de funcionamento o chaveamento de componentes eletrônicos, normalmente MOSFETs e transistores bipolares. Nele é possível variar a polaridade da tensão e sua magnitude, podendo assim determinar o sentido da corrente. É muito utilizada para controle de rotação de motores CC, e que será essa função em nosso projeto. Mas também é possível encontrar em outras aplicações como, conversores CA/CA, conversores CC/CC, inversores de potência, etc.

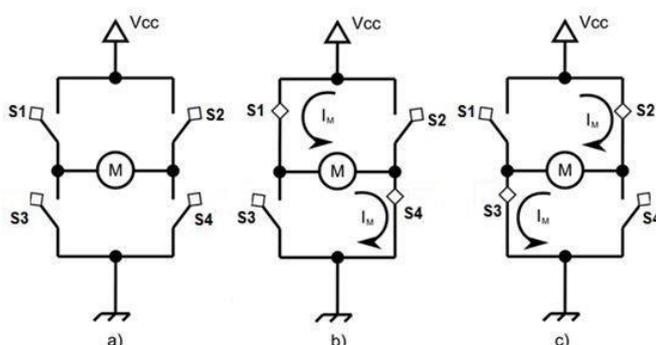


Figura 4 - Funcionamento da ponte H.

O circuito da Figura 4 nos mostra de modo simples como funciona uma ponte H. Nele se utiliza quatro chaves, representadas por (S1, S2, S3, S4), que são comutadas de forma alternada, (S1-S4) ou (S2-S3).

Acionando-se em conjunto, as chaves S1-S4, a corrente fluirá da direita para a esquerda no motor adotando um sentido. Do mesmo modo, acionando-se em conjunto, as chaves S2-S3, a corrente fluirá da esquerda para a direita no motor, adotando um sentido contrário. As chaves S1 e S2, assim como as chaves S3 e S4, não podem ser comutadas ao mesmo tempo, pois tal manobra fecharia um curto na fonte de alimentação. Na Figura 5, podemos ver um módulo com ponte H para Arduino.

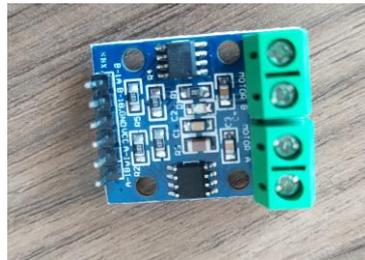


Figura 5 - Módulo ponte H L9110.

e) Motor CC com redução

O Motor CC com redução (Figura 6) é alimentado com corrente contínua (CC), isso permite que sua velocidade seja controlada através da variação de tensão (CHAPMAN, 2013). Como esse motor tem redução, isso permite que ele seja menor e mais leve, a redução pode variar entre redutor de engrenagem, planetários sem-fim ou cônico.



Figura 6 - Motor CC com redução e roda de borracha.

f) Módulo JOYSTICK 3 eixos (controle)

O Módulo JOYSTICK (Figura 7) é um módulo direcional analógico. Sua função é mandar um sinal informando a direção desejada. Um JOYSTICK permite não só indicar direita, esquerda, para baixo e para cima, através dos eixos X e Y, mas também

indicar as diagonais esquerda superior, direita superior, esquerda inferior e direita inferior. Esse módulo também conta com um botão, se referindo ao eixo Z.

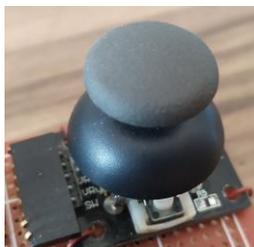


Figura 7 - Módulo JOYSTICK 3 eixos.

2.1 Onde Adquirir os Componentes

Os componentes podem ser adquiridos em lojas físicas e/ou virtuais. Procurando na internet é possível encontrar várias lojas e com uma grande variedade de componentes e preços.

Em Juiz de Fora, MG, é possível encontrar lojas físicas de componentes eletrônicos, onde conseguimos comprar alguns componentes gerais de uma forma mais rápida, como LEDs e resistores. Mas também há também loja do tipo física/virtual de componentes para Arduino.

Foi possível também localizar os componentes em sites internacionais como Aliexpress (ALIEXPRESS, 2020) e Wish (WISH, 2020), mas devido ao preço atual do dólar, acima dos R\$5,00 (10/09/2020), e o frete cobrado para cada componente, se torna inviável a compra nesses sites.

Lembrando que há diversos sites com diferentes preços e tempo de entrega. Por isso aconselho pesquisar bastante para encontrar a melhor opção, preço x tempo de entrega, de acordo com o seu projeto.

3 RESULTADOS E DISCUSSÃO

Nessa seção do artigo, o resultado do projeto será apresentado bem como a simulação individual e também completa para os módulos utilizados. Também, será abordado e discutido o preço para os componentes baseado em diferentes lojas pesquisadas.

Para cada componente simulado, terá um código de programação. Os códigos não são fixos, ou seja, poderá alterar de acordo com cada projeto. A seguir serão apresentados os principais módulos:

a) Arduino Uno

No Tinkercad conseguimos encontrar o Arduino UNO que será usado nas simulações, ver Figura 8. Lembrando que o mesmo possui 6 portas analógica e 14 portas digitais I/O (entrada e saída), sendo 6 do tipo PWM.

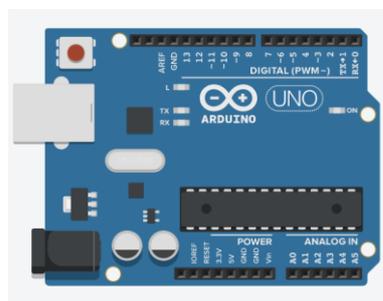


Figura 8 - Arduino Uno no Tinkercad.

b) Módulo Wireless NRF24L01

No TINKERCAD ainda não é possível simular o funcionamento do módulo Wireless NRF24L01(setembro/2020). Por isso, usamos o *software* Fritzing (FRITZING, 2020) para demonstrar a ligação e a própria IDE do Arduino para o código. O Fritzing é um programa, gratuito, onde é possível criar um circuito virtual do seu projeto e até transformá-lo em uma PCB. No *software* é possível encontrar vários componentes de Arduino e eletrônica em geral, como LED, resistor, diodo, etc.

A ligação desse módulo é a mesma tanto para o módulo emissor (controle), quanto para o receptor (robô), Figura 9. Os pinos MOSI, MISO, SCK deverão ser ligados nos pinos 11, 12 e 13 respectivamente. Já os pinos CE, CSN poderão ser ligados em qualquer um do tipo digital, lembrando só de declarar no seu código. Esse módulo pode ser alimentado de 1,9-3,6V, por isso, use a alimentação 3,3V do Arduino ou alguma alimentação externa dentro do limite do módulo.

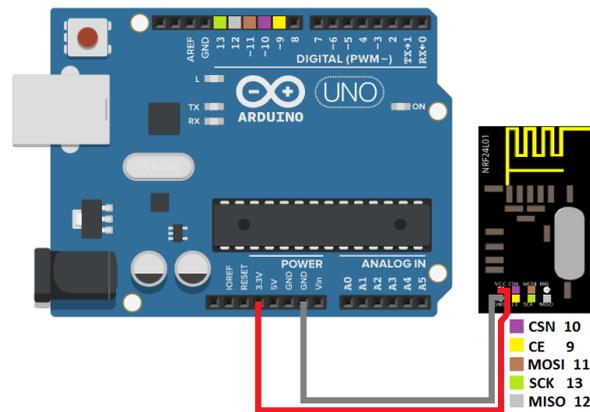


Figura 9 - Ligação Módulo Wireless NRF24L01 com Arduino UNO no Fritzing.

Para o uso do módulo é necessário baixar e instalar a biblioteca do mesmo. Para *download* dessa biblioteca basta pesquisar “biblioteca NRF24”, que é possível localizar inúmeros sites contendo a mesma, e muitos com tutorial de como instalá-la, caso tenha alguma dúvida.

Na Figura 10, podemos ver o exemplo de dois códigos utilizados para simular o controle e o robô, onde o módulo emissor mesmo envia um sinal para o módulo receptor.

ENVIA

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(9, 10);
const byte address[6] = "00001"

void setup(){
  radio.begin();
  radio.openWritingPipe(address)
  radio.setPALevel(RF24_PA_HIGH)
  radio.stopListening();
}

void loop(){
  int enviar = 1;
  radio.write(&enviar, sizeof(enviar));
  delay(100);
}
```

RECEBE

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(9, 10);
const byte address[6] = "00001";

void setup(){
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_HIGH);
  radio.startListening();
}

void loop() {
  if(radio.available()){
    int receber;
    radio.read(&receber, sizeof(receber));
    Serial.println(receber);
  }
}
```

Figura 10 - Código Módulo Wireless NRF24L01.

c) Motor CC com redução

No Tinkercad é possível encontrar o motor CC com redução e simulá-lo juntamente com o Arduino. Para essa simulação o motor foi ligado da seguinte forma, conforme podemos ver na Figura 11: O polo negativo do motor no pino GND do Arduino e o polo positivo na porta 3 do Arduino. Lembrando que para controlar o motor, o polo positivo pode ser ligado em qualquer porta PWM, sendo necessário declarar a mesma no código. Para uma maior velocidade é recomendado utilizar uma fonte externa para os motores, pois o Arduino é capaz de fornecer apenas 40 mA por porta e o motor consome cerca de 200 mA.

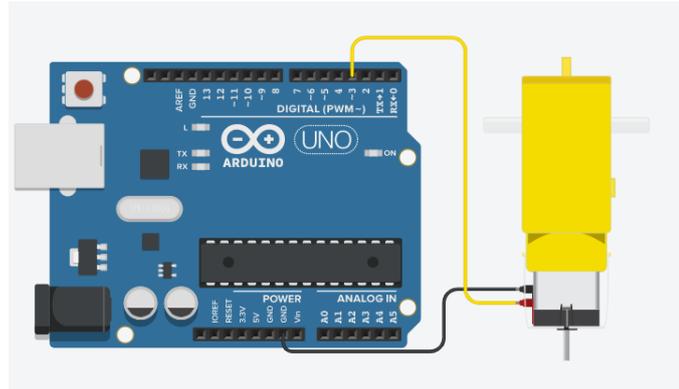


Figura 11 - Ligação motor CC com Arduino UNO no Tinkercad.

A Figura 12 apresenta um exemplo simples de código variando a velocidade dos motores com a função `analogWrite`.

```

const int motor = 3;           //pino motor no Arduino

void setup(){
  pinMode(motor, OUTPUT);     //pino do motor como saída
}

void loop(){
  analogWrite(motor, 255);     //velocidade máxima motor
  delay(2000);                 //espera 2s
  analogWrite(motor, 127);     //metade da velocidade motor
  delay(2000);                 //espera 2s
  analogWrite(motor, 0);       //motor parado
  delay(2000);                 //espera 2s
}

```

Figura 12 - Código motor CC

d) Ponte H

Para efeito de simulação vamos utilizar o CI ponte H - L293D, Figura 13, pois não é possível encontrar o módulo ponte H no Tinkercad. Em comparação com o módulo ponte H, o CI teremos que fazer mais ligações, gastando assim mais jumpers e ocupando mais portas no Arduino. No esquemático abaixo utilizamos também uma alimentação externa para os motores.

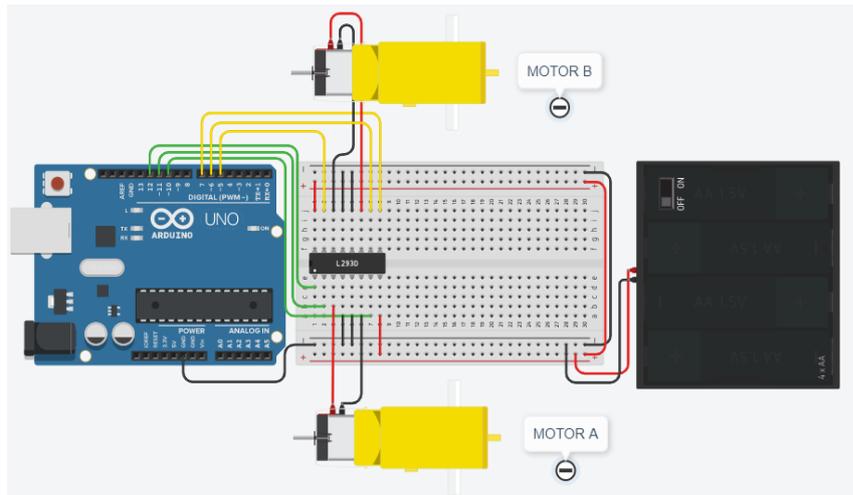


Figura 13 - Ligação motor CC e ponte H com Arduino UNO no Tinkercad.

Na Figura 14, é possível ver que foram criadas funções que chamamos de “frente()”, “re()” e “parar()”, com a finalidade de organização, para a função principal “void loop()” não ficar muito cheia. Para chamar as funções criadas na função principal, é necessário apenas “chamá-la” com o mesmo nome dado.

```

const int motorA1 = 10; //pino sentido 1 motor A arduino
const int motorA2 = 11; //pino sentido 2 motor A arduino
const int ativa_A = 12; //pino ativa motor A arduino

const int motorB1 = 5; //pino sentido 1 motor B arduino
const int motorB2 = 6; //pino sentido 2 motor B arduino
const int ativa_B = 7; //pino ativa motor B arduino

void setup(){
  pinMode(ativa_A, OUTPUT); //pino como saída
  pinMode(motorA1, OUTPUT); //pino motor A como saída
  pinMode(motorA2, OUTPUT); //pino motor A como saída
  digitalWrite(ativa_A, HIGH); //ativa motor A

  pinMode(ativa_B, OUTPUT); //pino como saída
  pinMode(motorB1, OUTPUT); //pino motor B como saída
  pinMode(motorB2, OUTPUT); //pino motor B como saída
  digitalWrite(ativa_B, HIGH); //ativa motor B
}

void loop(){
  frente(); //chamando a função criada "frente"
  delay(2000); //espera 2s

  parar(); //chamando a função criada "parar"
  delay(1000); //espera 1s

  re(); //chamando a função criada "re"
  delay(2000); //espera 2s

  parar(); //chamando a função criada "parar"
  delay(1000); //espera 1s
}

void frente(){ //função motores sentido 1
  analogWrite(motorA1, 0);
  analogWrite(motorA2, 255); //motor A velocidade máxima
  analogWrite(motorB1, 0);
  analogWrite(motorB2, 255); //motor B velocidade máxima
}

void re(){ //função motores sentido 2
  analogWrite(motorA1, 255); //motor A velocidade máxima
  analogWrite(motorA2, 0);
  analogWrite(motorB1, 255); //motor B velocidade máxima
  analogWrite(motorB2, 0);
}

void parar(){ //função motores pararem
  analogWrite(motorA1, 0);
  analogWrite(motorA2, 0);
  analogWrite(motorB1, 0);
  analogWrite(motorB2, 0);
}

```

Figura 14 - Código motor CC com ponte H.

e) Módulo Joystick 3 eixos

Como no módulo Wireless NRF24L01, ainda não é possível simular no Tinkercad o funcionamento do módulo Joystick (setembro/2020). Por isso, para esse caso, também utilizamos o *software* Fritzing para demonstrar a ligação e a própria IDE do Arduino para o código.

Nesse módulo, conseguimos ler 3 eixos, X, Y e Z, sendo que os eixos X e Y são analógicos, com leitura de 0 a 1023, e o Z digital, leitura de 0 ou 1. No exemplo, Figura 15, utilizamos as portas A0 e A1, para os eixos X e Y respectivamente e a porta digital 2, para o eixo Z. Na Figura 16, podemos ver seu código.

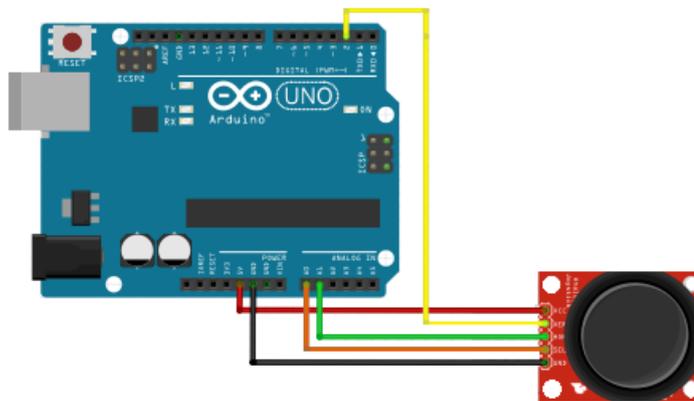


Figura 15 - Ligação módulo JOYSTICK com Arduino UNO no Fritzing.

```
int pino_x = A0; //leitura do eixo X na porta A0
int pino_y = A1; //leitura do eixo Y na porta A0
int pino_z = 2; //leitura do eixo X na porta 2

int val_x; //variável para receber o eixo X
int val_y; //variável para receber o eixo Y
int val_z; //variável para receber o eixo Z

void setup(){
  pinMode(pino_z, INPUT); //declarando o JOYSTICK como entrada
}

void loop(){
  val_x = analogRead(pino_x); //lendo o valor do eixo X
  val_y = analogRead(pino_y); //lendo o valor do eixo Y
  val_z = digitalRead(pino_z); //lendo o valor do eixo Z
  delay(30); //espera de 30ms para a próxima leitura
}
```

Figura 16 - Código módulo JOYSTICK.

f) Esquemático completo

Tendo o conhecimento dos componentes utilizados tanto para o controle, quanto para o robô, chegamos ao esquemático completo, como podemos ver a seguir nas Figuras 17 e 19, e suas respectivos códigos, Figuras 18 e 20.

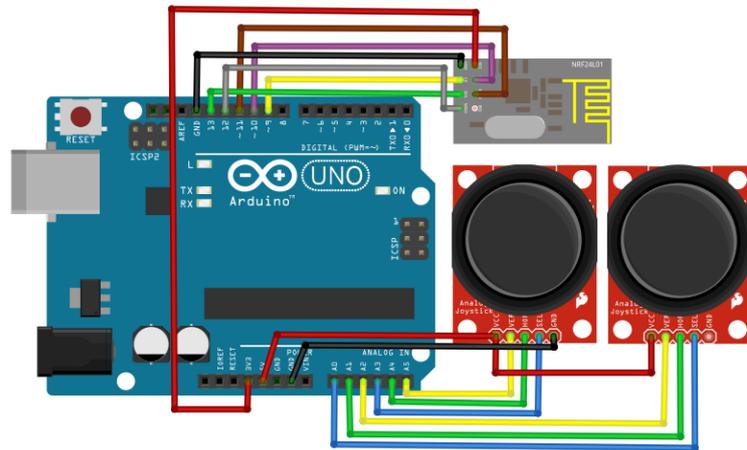


Figura 17 - Ligação completa controle no Fritzing.

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(9, 10);
const byte address[6] = "00001";

//JOYSTICK 1 - ESQUERDA - ACELERAR
const int pino_x1 = A5;
const int pino_y1 = A4;
const int pino_z1 = A3;

//JOYSTICK 2 - DIREITA - VIRAR
const int pino_x2 = A2;
const int pino_y2 = A1;
const int pino_z2 = A0;

struct Data_Package{
  byte x1;
  byte x2;
  byte y1;
  byte y2;
  byte z1;
  byte z2;
};
Data_Package data;

void setup(){
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_HIGH);
  radio.stopListening();

  pinMode(pino_z1, INPUT_PULLUP);
  pinMode(pino_z2, INPUT_PULLUP);
}

void loop(){
  data.x1 = map(analogRead(pino_x1), 0, 1023, 0, 10);
  data.y1 = map(analogRead(pino_y1), 0, 1023, 10, 0);
  data.z1 = (digitalRead(pino_z1)*-1)+1;

  data.x2 = map(analogRead(pino_x2), 0, 1023, 10, 0);
  data.y2 = map(analogRead(pino_y2), 0, 1023, 0, 10);
  data.z2 = (digitalRead(pino_z2)*-1)+1;

  radio.write(&data, sizeof(Data_Package));
  delay(20);
}
```

Figura 18 - Código do controle.

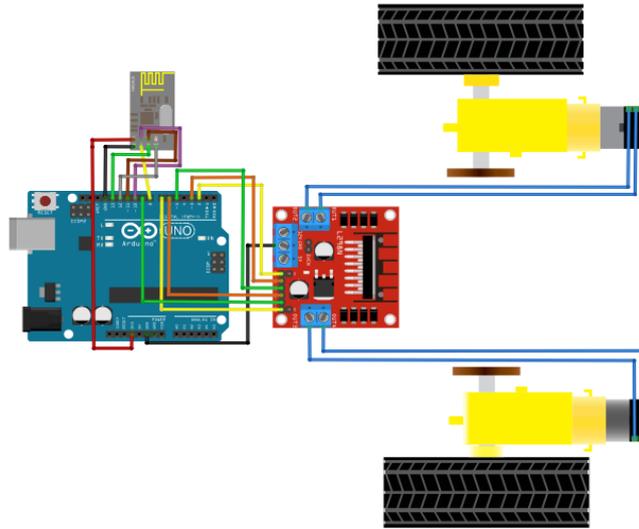


Figura 19 - Ligação completa robô no Fritzing.

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
RF24 radio(9, 10);
const byte address[6] = "00001";

struct Data_Package{
  byte x1;
  byte x2;
  byte y1;
  byte y2;
  byte z1;
  byte z2;
};
Data_Package data;

int AIA = 3;
int AIB = 5;
int BIA = 9;
int BIB = 6;
int VELOCIDADE;

void setup(){
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.disableCRC();
  radio.setPALevel(RF24_PA_MAX);
  radio.startListening();

  pinMode(AIA, OUTPUT); pinMode(AIB, OUTPUT);
  pinMode(BIA, OUTPUT); pinMode(BIB, OUTPUT);
}

void loop(){
  if(radio.available()){
    radio.read(&data, sizeof(Data_Package));

    if(data.y1 >= 7 && data.y1 <= 10){ frente(); }
    else if(data.y1 > 3 && data.y1 < 7){ parar(); }
    else if(data.y1 >= 0 && data.y1 <= 3){ re(); }

    if(data.x2 >= 7 && data.x2 <= 10){ direita(); }
    else if(data.x2 >= 0 && data.x2 <= 3){ esquerda(); }
  }

  void parar(){
    analogWrite(AIA, 0); analogWrite(AIB, 0);
    analogWrite(BIA, 0); analogWrite(BIB, 0);
  }
  void frente(){
    analogWrite(AIA, 255); analogWrite(AIB, 0);
    analogWrite(BIA, 255); analogWrite(BIB, 0);
  }
  void re(){
    analogWrite(AIA, 0); analogWrite(AIB, 255);
    analogWrite(BIA, 0); analogWrite(BIB, 255);
  }
  void direita(){
    analogWrite(AIA, 0); analogWrite(AIB, 0);
    analogWrite(BIA, 255); analogWrite(BIB, 0);
  }
  void esquerda(){
    analogWrite(AIA, 255); analogWrite(AIB, 0);
    analogWrite(BIA, 0); analogWrite(BIB, 0);
  }
}

```

Figura 20 - Código do robô.

Após pesquisa em 3 fornecedores, LOJA 1 (FILIPEFLOP, 2020), LOJA 2 (CURTO CIRCUITO, 2020) e LOJA 3 (VAMUINO, 2020) chegamos a opções bem diversificadas quanto ao planejamento escolhido para o projeto, vide Tabela 2 e Tabela 3. Na LOJA 1, há uma maior diversidade de componentes, mais tempo no mercado e os seus componentes tem um tutorial de como utilizá-los, mas com um preço mais elevado. A LOJA 2, tem uma grande diversidade de componentes e um preço/custo total menor. Já a LOJA 3, o tempo de entrega para nosso projeto se dá de uma forma mais rápida, conseguindo retirar no mesmo dia.

Tabela 2 - Preço dos componentes do robô em 3 lojas.

COMPONENTE	LOJA 1	LOJA 2	LOJA 3
Arduino	R\$ 62,90	R\$ 36,60	R\$ 40,00
Rodas com caixa de redução (x2)	R\$ 43,60	R\$ 35,00	R\$ 36,00
Módulo Ponte H L298N	R\$ 22,90	R\$ 17,00	R\$ 20,00
Protoboard	R\$ 16,90	R\$ 11,95	R\$ 15,00
Módulo Wireless NRF24L01	R\$ 34,90	R\$ 8,60	R\$ 10,00
TOTAL	R\$ 181,20	R\$ 109,15	R\$ 121,00

Tabela 3 - Preço dos componentes do controle em 3 lojas.

COMPONENTE	LOJA 1	LOJA 2	LOJA 3
Arduino	R\$ 62,90	R\$ 36,60	R\$ 40,00
Módulo Joystick	R\$ 9,90	R\$ 9,25	R\$ 10,00
Chave liga/desliga	R\$ 3,90	R\$ 1,25	R\$ 2,00
Módulo Wireless NRF24L01	R\$ 34,90	R\$ 8,60	R\$ 10,00
TOTAL	R\$ 111,60	R\$ 55,70	R\$ 62,00

4 CONCLUSÃO

Este artigo abordou as etapas do processo de elaboração de um robô controlado remotamente através do Arduino bem como o orçamento do mesmo. Durante o projeto foram utilizados programas de simulação que auxiliaram nas escolhas dos componentes e no desenvolvimento dos *scripts* a serem utilizados.

Como base nos resultados obtidos, foi possível criar um tutorial com os componentes e códigos necessários, realizar um levantamento de preços em três lojas distintas e analisar qual delas oferece o melhor custo-benefício e um período de entrega viável à conclusão do projeto.

REFERÊNCIAS

ALIEXPRESS. Disponível em: <<https://pt.aliexpress.com/>>. Acesso em: 05 set. de 2020.

ANALOGWRITE, **analogWrite()**. Disponível em: <<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>>. Acesso em: 10 set. de 2020.

ARDUINO, **Arduino Uno**. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 01 set. de 2020.

CAMPOS, Roberto Augusto Freitas. **Automação Residencial utilizando Arduino e Aplicações Web**. 2014. 85 f. Trabalho de Conclusão de Curso (Engenharia de Computação)-Centro Universitário de Brasília, Brasília, 2014.

CHAPMAN, Stephen J. **Fundamentos de Máquinas Elétricas**. 5^a ed. Porto Alegre: AMGH, 2013.

COLARES, Richard A., LIMA, Lincoln F. Disponível em: <<https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/auxilio-a-deficientes-visuais>>. Acesso em: 10 out. de 2020.

CURTO CIRCUITO. Disponível em: <<https://www.curtocircuito.com.br/>>. Acesso em: 10 set. de 2020.

FILIFELOP. Disponível em: <<https://www.filipeflop.com/>>. Acesso em: 02 set. de 2020.

FRITZING. **Fachhochschule Potsdam**. Software. Disponível em: <<https://fritzing.org/>>. Acesso em: 09 de out. de 2020.

HIRZEL, Timothy. **PWM**, 2018. Disponível em: <<https://www.arduino.cc/en/Tutorial/PWM>>. Acesso em: 08 set. de 2020.

MONK, Simon. **30 Projetos com Arduino**. 2ª ed. Porto Alegre: Bookman, 2014

NORDIC, **nRF24 Series**, 2020. Disponível em: <<https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF24-series>>. Acesso em: 10 set. de 2020.

SOFTWARE, **Arduino**, 2020. Software. Disponível em: <<https://www.arduino.cc/en/Main/Software>>. Acesso em: 14 set. de 2020.

TINKERCAD. **AUTODESK TINKERCAD**, 2020. Software. Disponível em: <<https://www.tinkercad.com/>>. Acesso em: 28 set. de 2020.

VAMUINO. Disponível em: <<https://lojinha.vamuino.com.br/>>. Acesso em: 05 set. de 2020.

WISH. Disponível em: <<https://www.wish.com/>>. Acesso em: 29 set. de 2020.