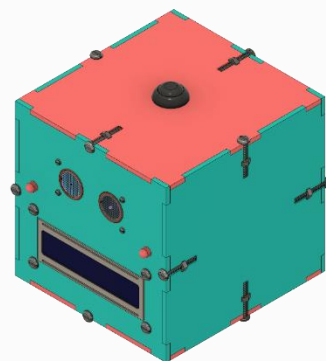
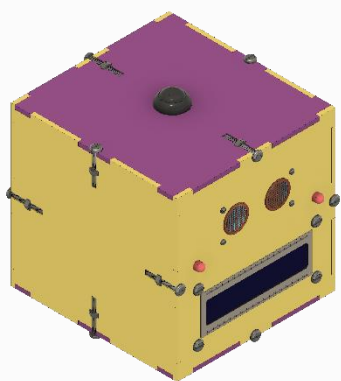


Construa o seu Robô Ludos

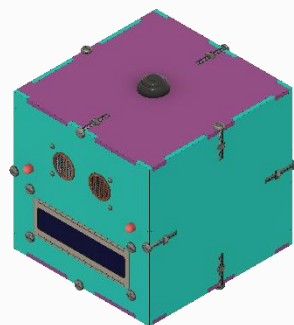
Aprenda a programação e eletrônica com Arduino ou outras plataformas através do Robô Ludos.



"Fácil montagem, divertido e lúdico para qualquer idade."



Praticidade, simplicidade e facilidade de uso em um único kit.



Escola Robô Lúdico

Muito obrigado e seja bem-vindo(a)!



Satisfação em tê-lo como um(a) amante do Robô Ludos. Sou o Diego Moreira e fascinado por trabalhar com o Arduino.

Eu trabalho na área de ensino há mais de 10 anos e tive experiência de trabalhar em diversos laboratórios, como: física, materiais de interesse na mecânica, sensores industriais e protótipos.

Além desses laboratórios, tive a oportunidade de ministrar aulas em escolas técnicas de Fortaleza-CE e, também, participar de alguns programas de ensino presencial e à distância do Governo Federal do Brasil: PRONATEC e ETEC.

Os laboratórios e a sala de aula me permitiram aprender bastante. Além disso, eu observei que era necessário criar metodologias com o intuito de facilitar e agilizar o aprendizado do aluno. O aluno precisa ter o contato com o conhecimento aplicado em situações reais.

Nesse período eu fiquei insatisfeito com o ensino em sala aula. Infelizmente, eu não tinha a possibilidade de mudar o sistema de ensino da escola em que eu trabalhava: um sistema ineficiente e com baixo investimento na aplicação de práticas para os alunos.

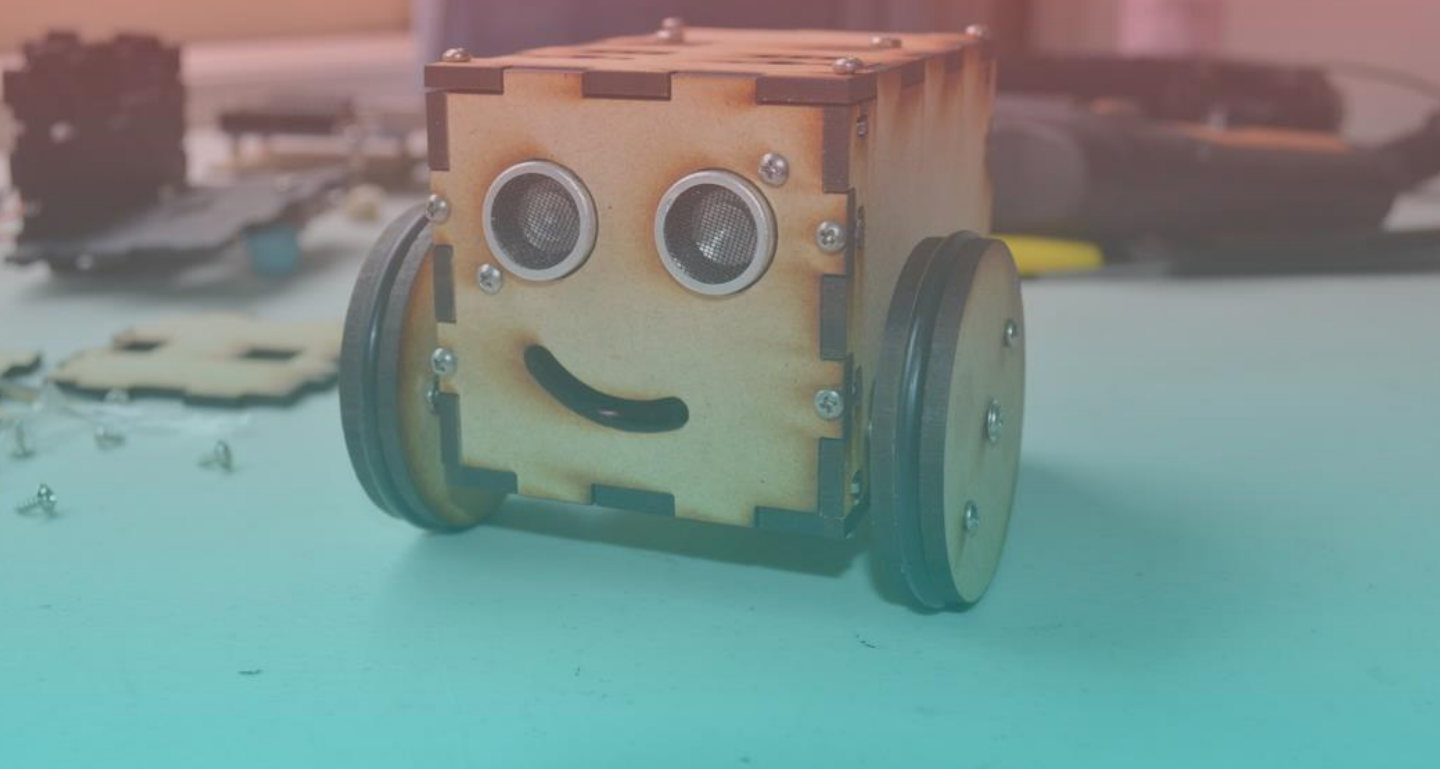
Foi nesse período que eu resolvi criar algumas metodologias para facilitar o ensino em sala de aula e anos depois surgiu a Robô Lúdico. Uma metodologia, uma escola que tem o intuito de facilitar o ensino de programação e robótica com Arduino e outras tecnologias.

Sumário

O Desafio no ensino da robótica com o Arduino	4
O que é o Robô Ludos	6
As características do Robô Ludos	8
Processo de Montagem do Robô Ludos	10
Hora da mão na massa	20
Documentação técnica do Robô Ludos	33
Suporte técnico e Grupo do WhatsApp do Robô Ludos.....	45

**“Se prepare!
A sua viagem pelo
mundo da robótica vai
começar agora.”**





O Desafio no ensino da Robótica com o Arduino

Eu tive muitos desafios quando comecei a aprender microcontroladores e eletrônica aplicada ao mundo da robótica com o Arduino.

Um desses desafios foi: **começar com alguma aplicação simples e com algo que eu pudesse ver funcionando na prática durante as minhas aulas de aprendizado.**

Eu observei isso não somente no meu aprendizado, mas também durante o momento em que eu fui para a sala de aula como professor, para ministrar aulas aos meus alunos.

Lá, eu percebi que outros professores também repetiam essa prática de deixar o aluno somente na bancada, programando um microcontrolador, fazendo leds piscarem e sensores detectando alguma coisa.

Entretanto, o aluno não conseguia visualizar aquilo tudo em uma situação prática.

Foi nessa época que eu concluí que isso era um problema e que era preciso criar um meio mais eficiente, mais interessante e amigável, para que o meu aluno pudesse aprender de forma prática.

Durante as aulas eu comecei a trabalhar e pensar em alguns kits, como por exemplo: **criar a estrutura de uma maquete de uma casa**, para fazer um sistema de automação residencial através do uso de microcontroladores ou do Arduino

Dessa forma, eu observei aumentou o envolvimento com a aula e meu aluno absorveu aquele conhecimento de **forma prática e através de uma aplicação real**. Eles conseguiram visualizar onde era possível aplicar os sensores e, também, **ser capaz de ter novas ideias, trabalhar a criatividade e criar novas soluções**.

De acordo com essas observações, eu criei o Robô Ludos.

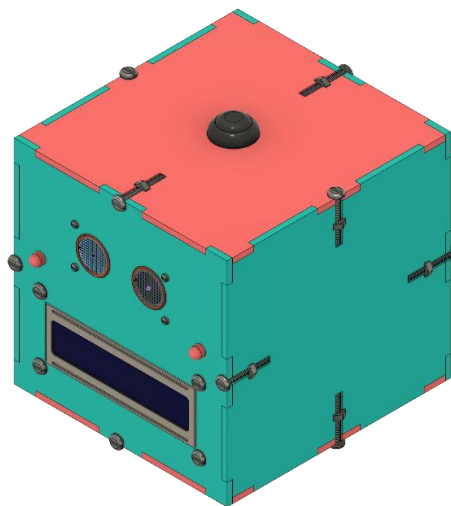


Figura 1 – Estrutura do Robô Ludos.

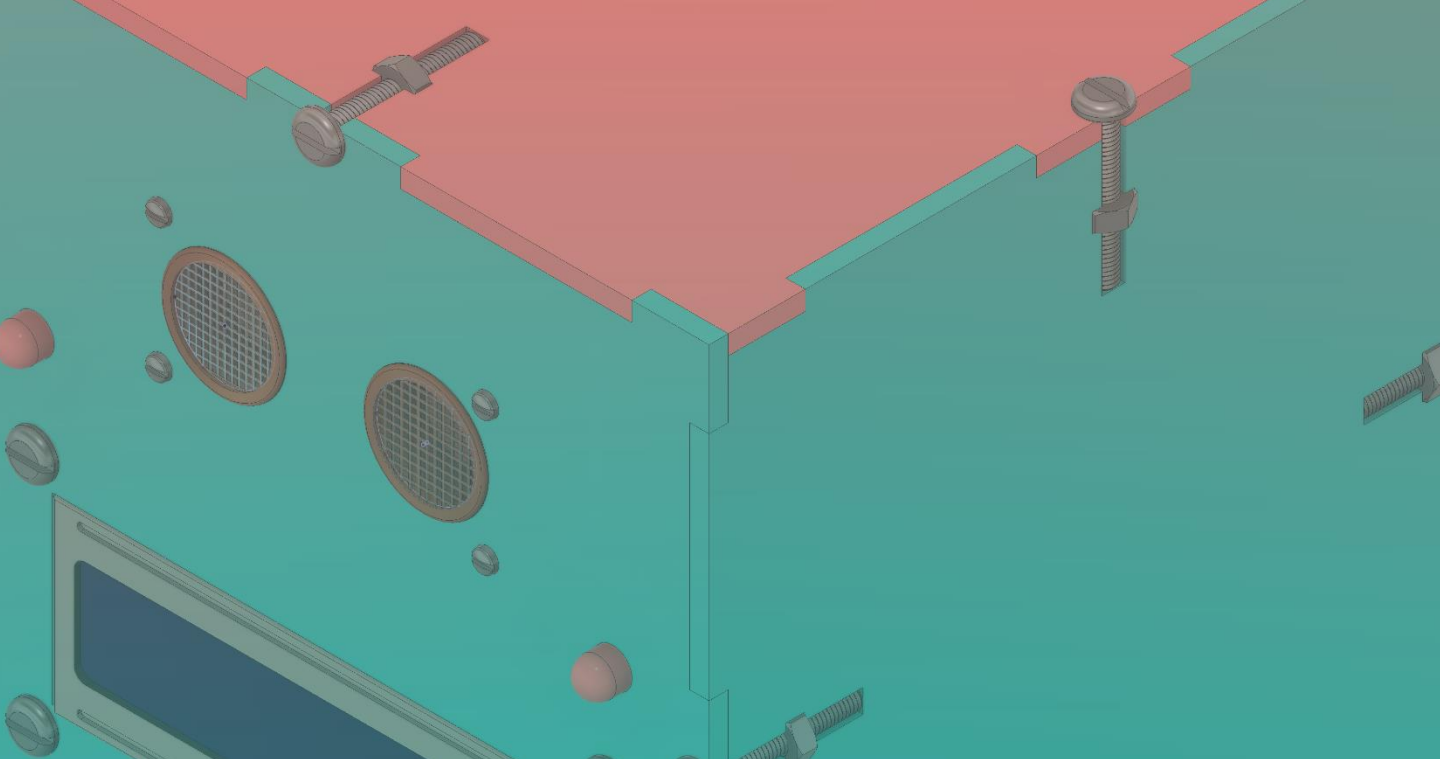
Ele foi criado com a finalidade de permitir com que você, que é iniciante ou professor e não tem um robô de fácil acesso, possa utilizar um **kit robótico simples**, com recursos variados e que ajudam você a tornar **sua aula mais didática e prática**.

Dessa forma, o Robô Ludos veio para **facilitar o aprendizado** e permitir que o aluno pegue o dispositivo robótico e **desenvolva vários projetos**.

Além disso, ele se propõe a:

- **facilitar os primeiros contatos com montagem do robô;**
- **permitir que seja uma aplicação simples e criativa.**

Tudo isso permite que seu aluno não tenha dificuldade inicial e que ele possa montar o seu primeiro kit de forma prática com o Arduino.



O que é o Robô Ludos ?

O Robô Ludos é um robô didático no formato de um cubo. Ele permite que você coloque qualquer tipo de Arduino internamente ou externamente, de modo a facilitar a programação e uso do seu kit.

Ele possui os seguintes componentes eletrônicos:

- 01 x Botão;
- 01 x Tela de LCD 16x2;
- 02 x LEDs Vermelhos;
- 01 x Sensor ultrassônico.

Além disso, você pode aproveitar a parte interna de aproximadamente 10 x 10 x 10 cm e fazer a instalação de outros sensores e módulos para trabalhar em conjunto com os outros componentes do robô.

A proposta é permitir com que você faça uma mistura de vários módulos e construa novas experiências com o seu robô.

Existem muitas ideias interessantes que são possíveis de serem construídas através de sua estrutura simples, fácil de ser montada e com um preço acessível com corte à laser ou impressão 3D.

Para que você tenha noção de custos, o valor do conjunto de peças cortadas a laser foi de R\$10.

Com R\$10 você consegue montar a estrutura em MDF e, em seguida, conectar os componentes de LCD, sensores e outras coisas.

Agora, você vai conhecer as características do robô Ludos.

As características do Robô Ludos

Quais são as características do robô Ludos? O que eu posso fazer com ele?

Bem, conforme já citei...

“O Robô Ludos foi desenvolvido para permitir com que o aluno e o professor tenham a primeira experiência com a robótica de forma acessível, simples e prática, para que o aluno possa pegar, manusear e montar o seu próprio robô.”

O Robô Ludos possui os seguintes elementos: Tela de Display LCD 16x2, LEDs, Sensor ultrassônico e botão.

Nós temos um total de 5 elementos. Todos eles estarão conectados em um circuito com o Arduino e permitirão que você desenvolva práticas em laboratório e em bancada.

Cada componente eletrônico tem suas próprias características de funcionamento.

O LCD, por exemplo, representa a boca do robô Ludos.

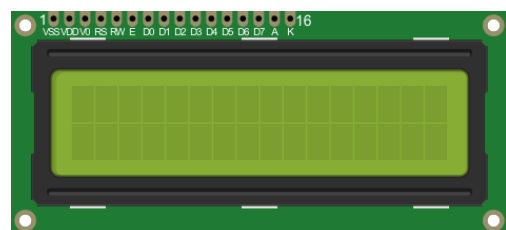


Figura 2 – Estrutura do Display de LCD.

Ele tem a finalidade de apresentar informações para o usuário. Essa informação, por exemplo, pode ser uma distância, uma contagem, uma mensagem de texto e muito mais..

Além do LCD, temos o sensor ultrassônico. Ele é utilizado para calcular distância.

Dessa forma, caso um objeto se aproxime, você pode programar para que essa distância seja apresentada no display de LCD.



Figura 3 – Sensor Ultrassônico.

Além disso, você pode gerar um alerta e informar que tem algo muito próximo do robô e fazer com que ele acione os LEDs como sinal de alerta.

Uma outra atividade interessante é a régua ultrassônica. Você posiciona o robô em um local, aponta para um ponto e aperta o botão que está sobre a cabeça do robô. Em seguida, ele irá calcular essa distância e mostrar o valor na tela do LCD.

Tudo isso e muito mais pode ser desenvolvido com o Robô Ludos, já que você pode adicionar outros elementos e criar novas ideias.

Por fim, temos dois LEDs de 5mm instalados na estrutura do nosso robô e que serão conectados aos pinos digitais do nosso Arduino.

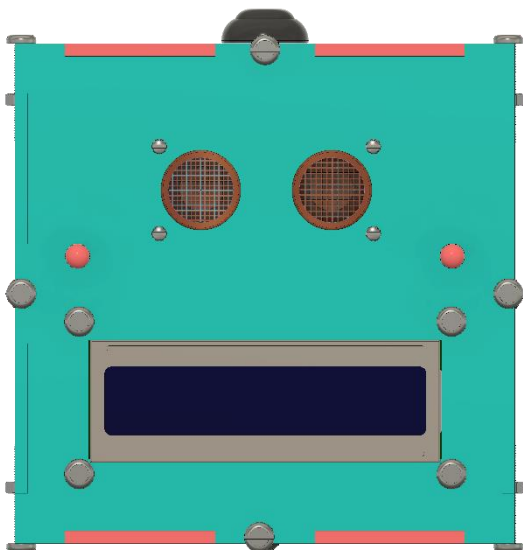


Figura 4 – Visualização frontal do Robô Ludos.

Agora que você compreendeu as características do Robô Ludos e sabe que ele é bastante espaçoso, você pode, também, instalar outros módulos internamente e criar novas funcionalidades para o seu robô.

Imagine que você quer colocar o seu robô para funcionar como um despertador.

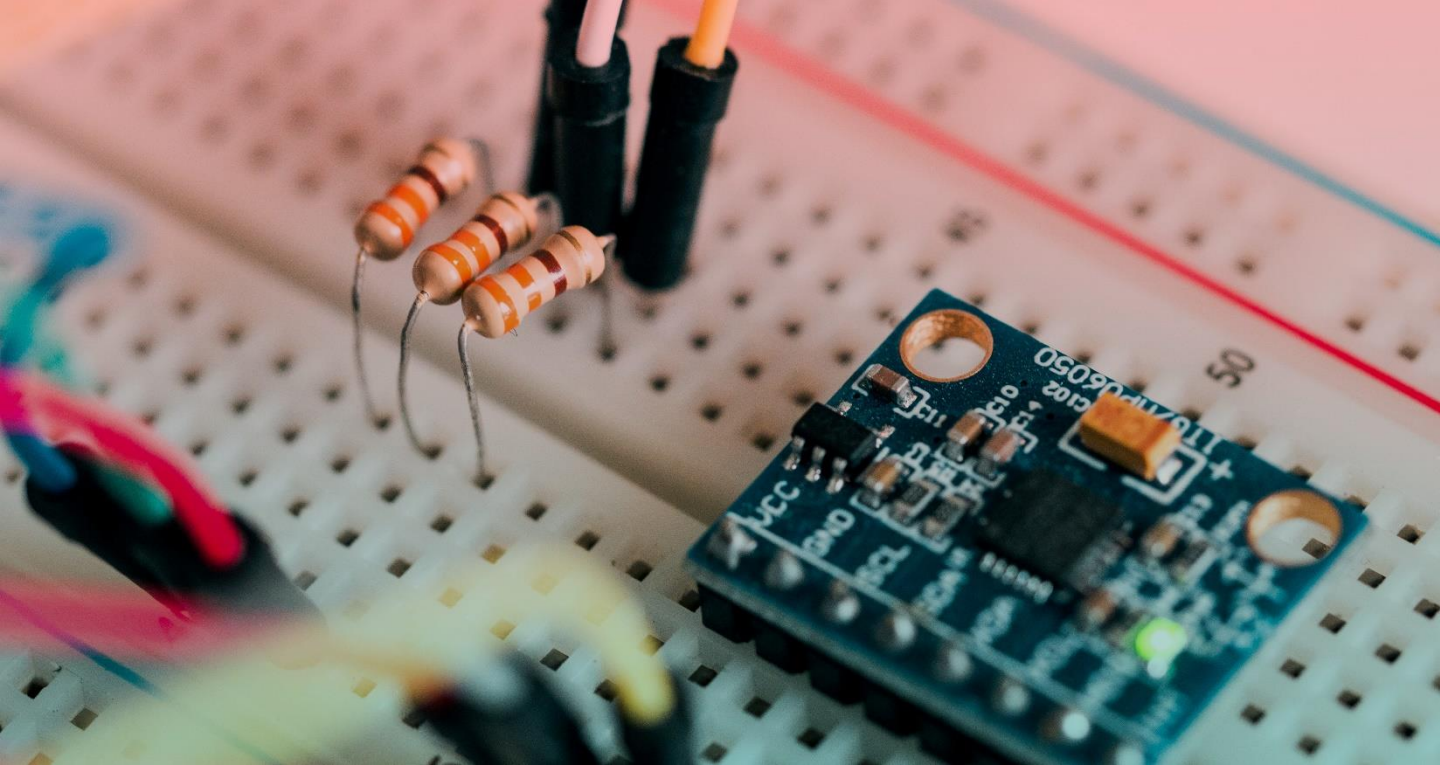
Como podemos fazer isso?

O robô Ludos não tem um módulo RTC em sua estrutura, mas você pode colocar um módulo RTC e um buzzer para que ele gere um alarme no horário desejado.

Observe que há muitas funcionalidades que você pode adicionar para criar novas características para o seu robô.

Com isso, você aprender de forma divertida, fácil e direcionada a diversas ideias criativas.

Agora, vamos falar sobre o processo de montagem da estrutura do Robô Ludos.



Processo de Montagem do Robô Ludos

O Robô Ludos possui o formato de um cubo e é representado pelas 6 faces apresentadas a seguir.

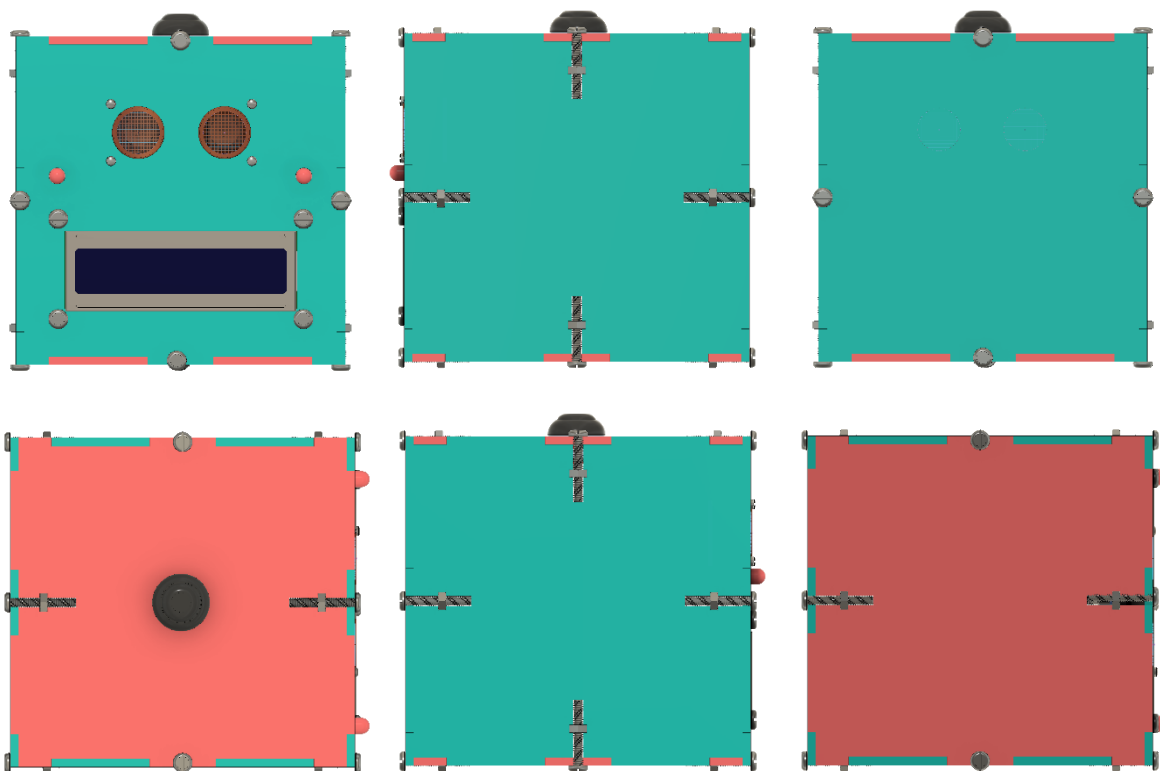


Figura 5 – Visualização de todas as faces do Robô Ludos.

Agora, vamos analisar cada estrutura que forma o robô Ludos.

Observe a face frontal do robô e veja os furos de fixação do LCD, do sensor ultrassônico e dos LEDs.

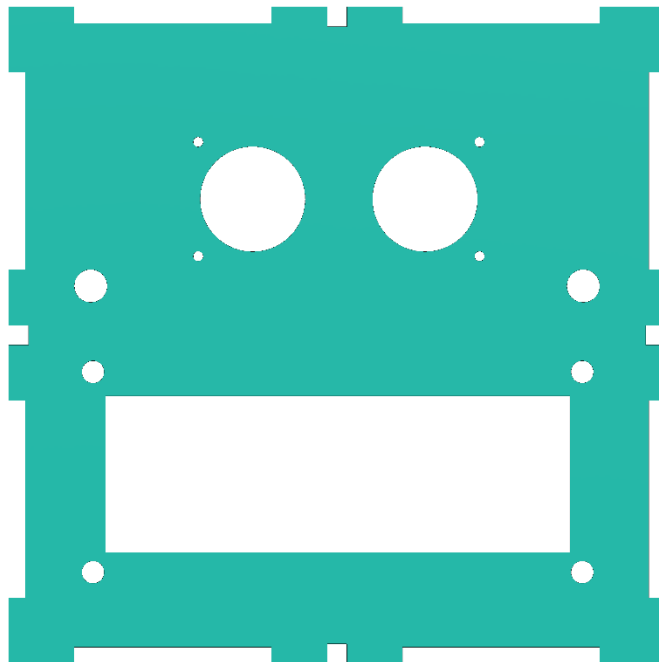


Figura 6 – Estrutura da face frontal do Robô Ludos.

A seguir, temos a face lateral esquerda e a face lateral direita. Observe a imagem e veja que elas possuem o mesmo formato.



Figura 7 – Tampas das laterais esquerda e direita do Robô Ludos.

Além disso, nós temos a tampa superior e a tampa inferior, que possuem, também, o mesmo formato.

A única diferença é que a tampa superior possui um furo de 12mm para fixar o botão, que poderá ser conectado em um pino digital do Arduino.

Com esse botão você pode fazer diversos tipos de experiências.



Figura 8 – Tampa superior (com furo) e inferior do Robô Ludos.

O botão utilizado é apresentado na figura a seguir. Ele é impermeável e possui uma rosca de 12 mm. Sua estrutura e seus detalhes técnicos são apresentados na figura abaixo.

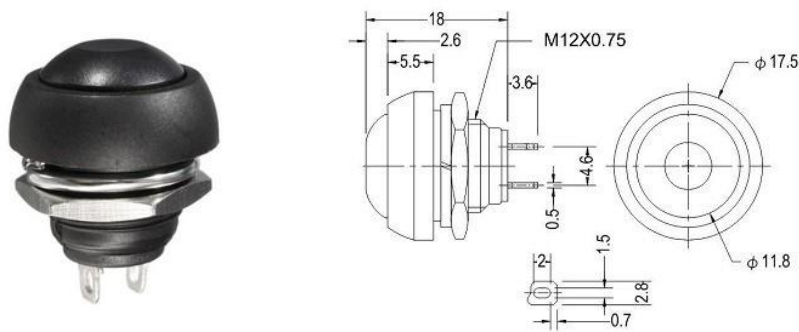


Figura 9 – Estrutura do Botão e suas dimensões.

Por fim, nós temos a face traseira, que possui o mesmo formato da face frontal. Essa estrutura tem um corte para você passar alguns cabos, caso você queira fazer a conexão através de uma placa de Arduino externamente ao robô.

Veja a estrutura na próxima página.

Essa é a estrutura da parte traseira do Robô Ludos.



Figura 10 – Estrutura da face frontal e traseira com o mesmo perfil de corte.

Além disso, há uma peça para melhorar o acabamento e criar um relevo do furo traseiro. Essa peça deve ser colada na estrutura e é apresentada a seguir.



Figura 11 – Peça para criação do relevo no corte da peça traseira do Robô Ludos.

Essas são as estruturas que formam as seis faces do robô e, além disso, a gente deve prestar atenção na forma de união dessas 6 faces.

As seis faces são unidas através de duas coisas importantes: as juntas de dedos e parafusos de fixação.

O início da montagem acontece através da fixação do sensor ultrassônico, LEDs e display de LCD na face do robô.

Instale o sensor ultrassônico e faça a fixação do parafuso M1.6 com sua respectiva rosca. O resultado da fixação é apresentado na figura abaixo.

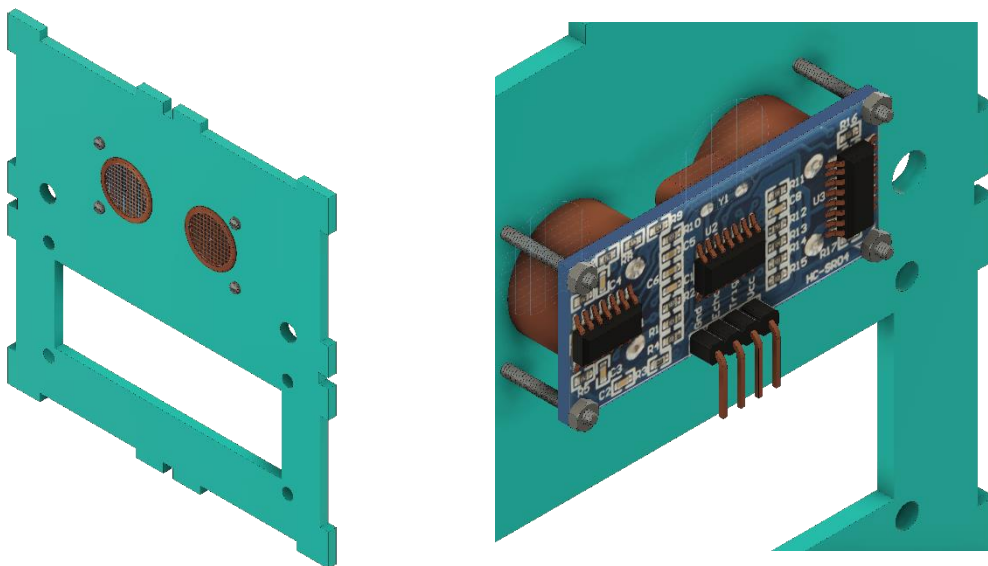


Figura 12 – Resultado da fixação do sensor ultrassônico na face frontal do Robô Ludos.

Após isso, instale os dois LEDs nos furos de 5mm. Utilize um pouco de cola quente na parte interna, para facilitar a fixação na estrutura.

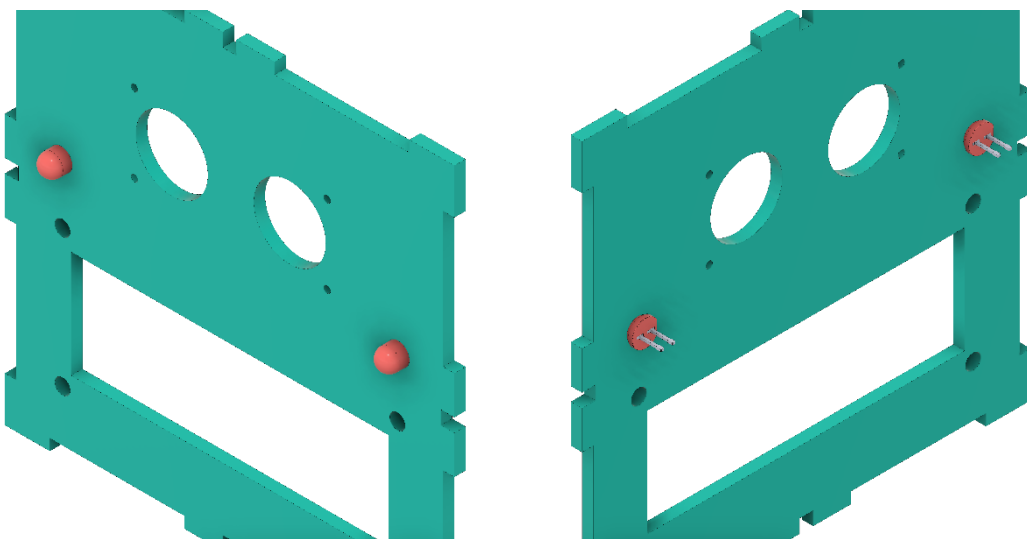


Figura 13 – Fixação dos LEDs na face frontal do Robô Ludos.

Antes de você realizar a conexão dos LEDs, sugerimos soldar fios nos terminais dos LEDs. Eles permitirão alongar os terminais dos LEDs e facilitar a conexão com os pinos do Arduino.

Finalmente, temos a conexão do LCD. Instale o LCD no local apresentado e faça a fixação dos parafusos M3 com sua rosca. O resultado é apresentado na figura abaixo.

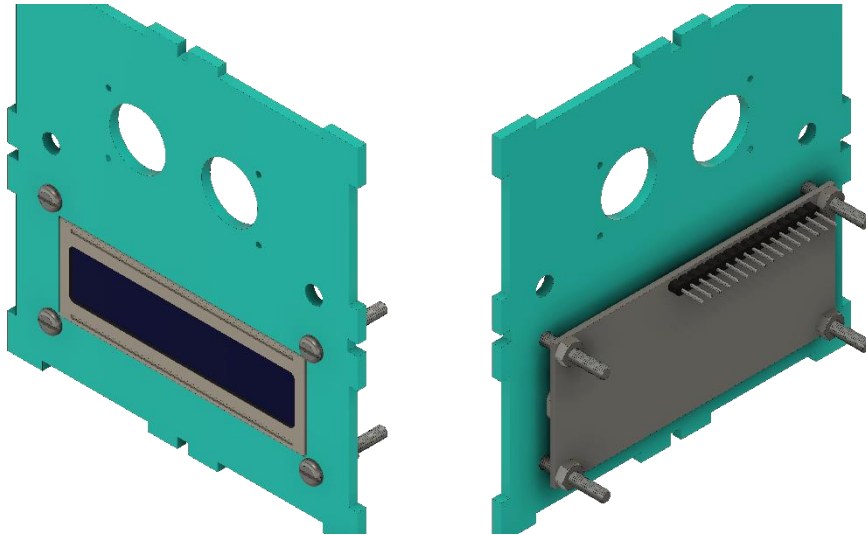


Figura 14 – Resultado da fixação do Display de LCD na face frontal do Robô Ludos.

Agora, você deve fazer a conexão do botão na tampa superior.

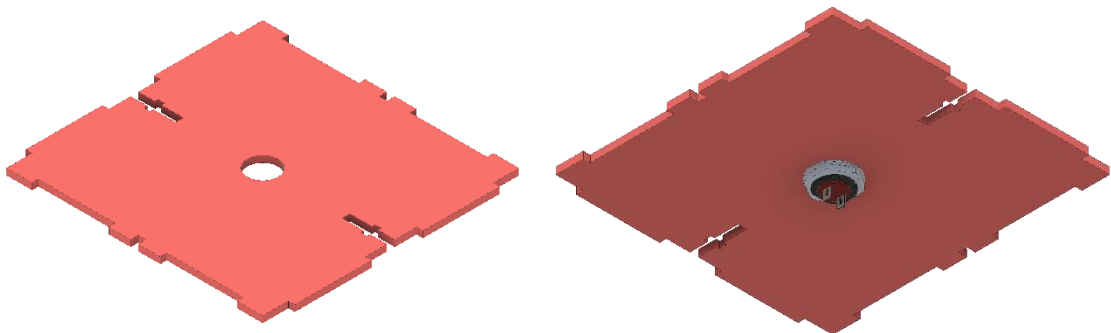


Figura 15 – Resultado da fixação do botão no furo da tampa superior.

Conforme é possível visualizar na figura acima, o botão é conectado através de sua rosca e uma porca de fixação.

Após a fixação dos componentes na estrutura do robô, chegou a hora em que você deve unir todas as partes do robô através dos parafusos de fixação M3.

O primeiro passo é realizar a fixação da parte frontal com a lateral direita através do parafuso M3 e sua rosca de trava. O resultado é apresentado abaixo.

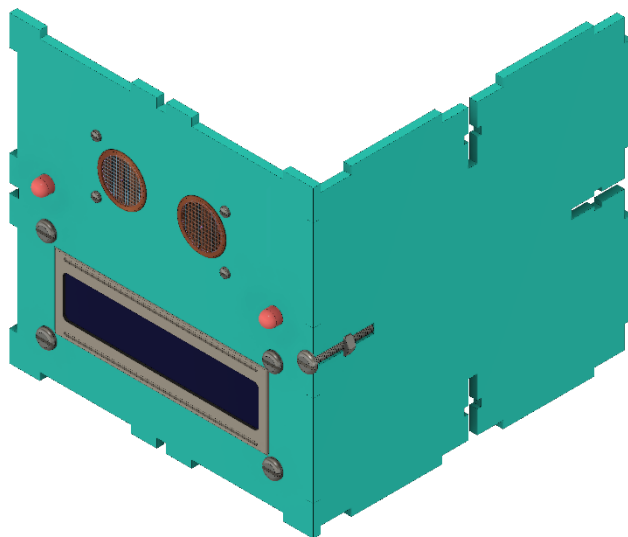


Figura 16 – Encaixe das face frontal com a face lateral direita do Robô Ludos.

O segundo passo consiste em fixar a tampa superior e fazer a fixação com os parafusos M3.

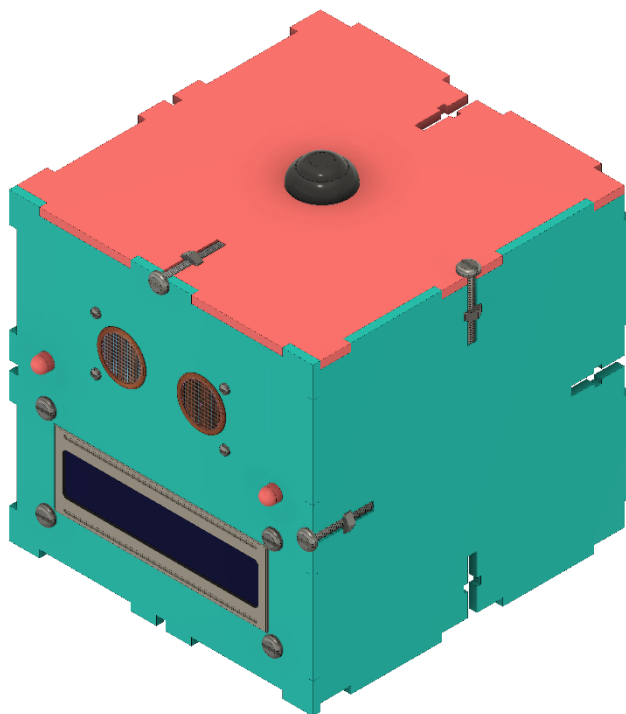


Figura 17 – Encaixe da tampa superior do Robô Ludos.

A seguir, continuaremos a montagem das outras faces.

No terceiro passo realize a fixação da tampa traseira com a parte que já foi montada anteriormente. Depois disso, cole a peça de relevo no furo. O resultado é apresentado na figura abaixo.

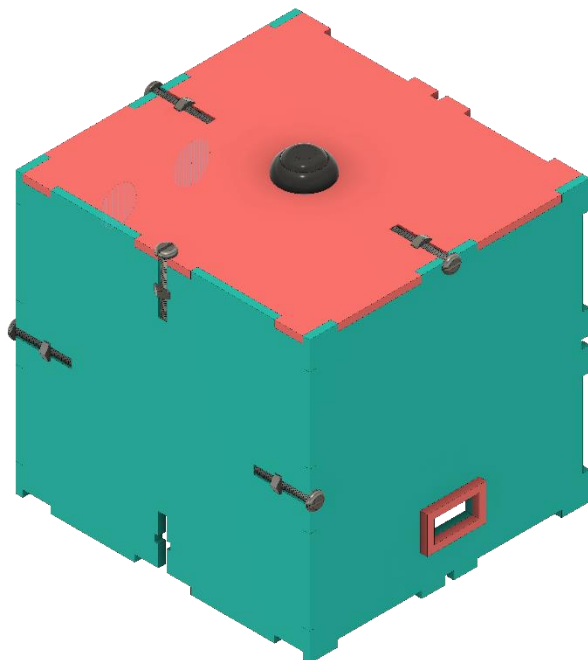


Figura 18 – Encaixe da tampa traseira e fixação do relevo no furo.

No quarto passo você deve fixar a lateral esquerda da face do seu robô. O resultado é apresentado na figura abaixo.

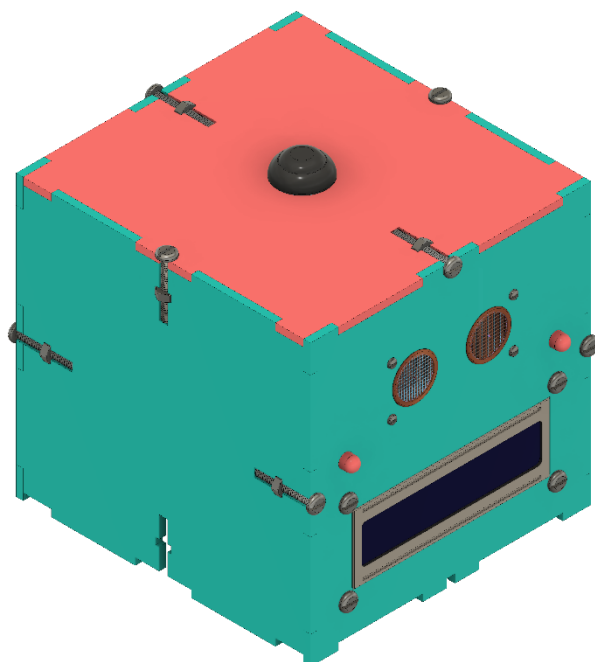


Figura 19 – Encaixe da tampa da lateral esquerda do Robô Ludos.

Por fim, temos a montagem da última parte do Robô Ludos.

Realize a fixação da tampa inferior do robô Ludos. O resultado é apresentado na figura abaixo.

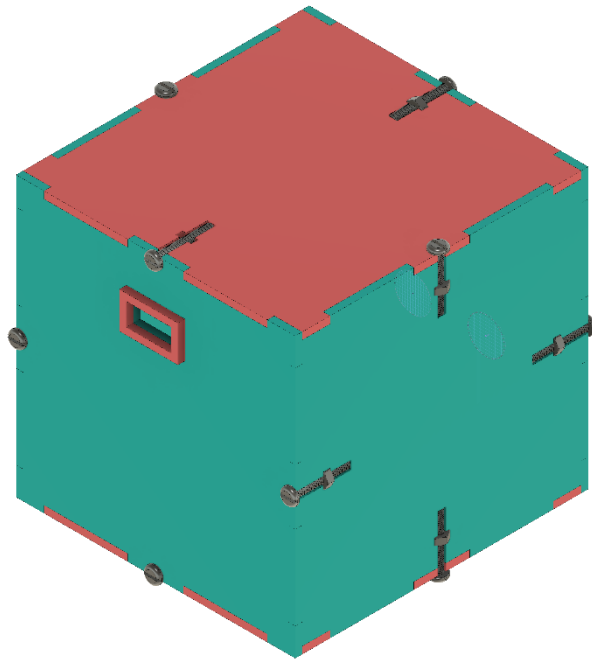
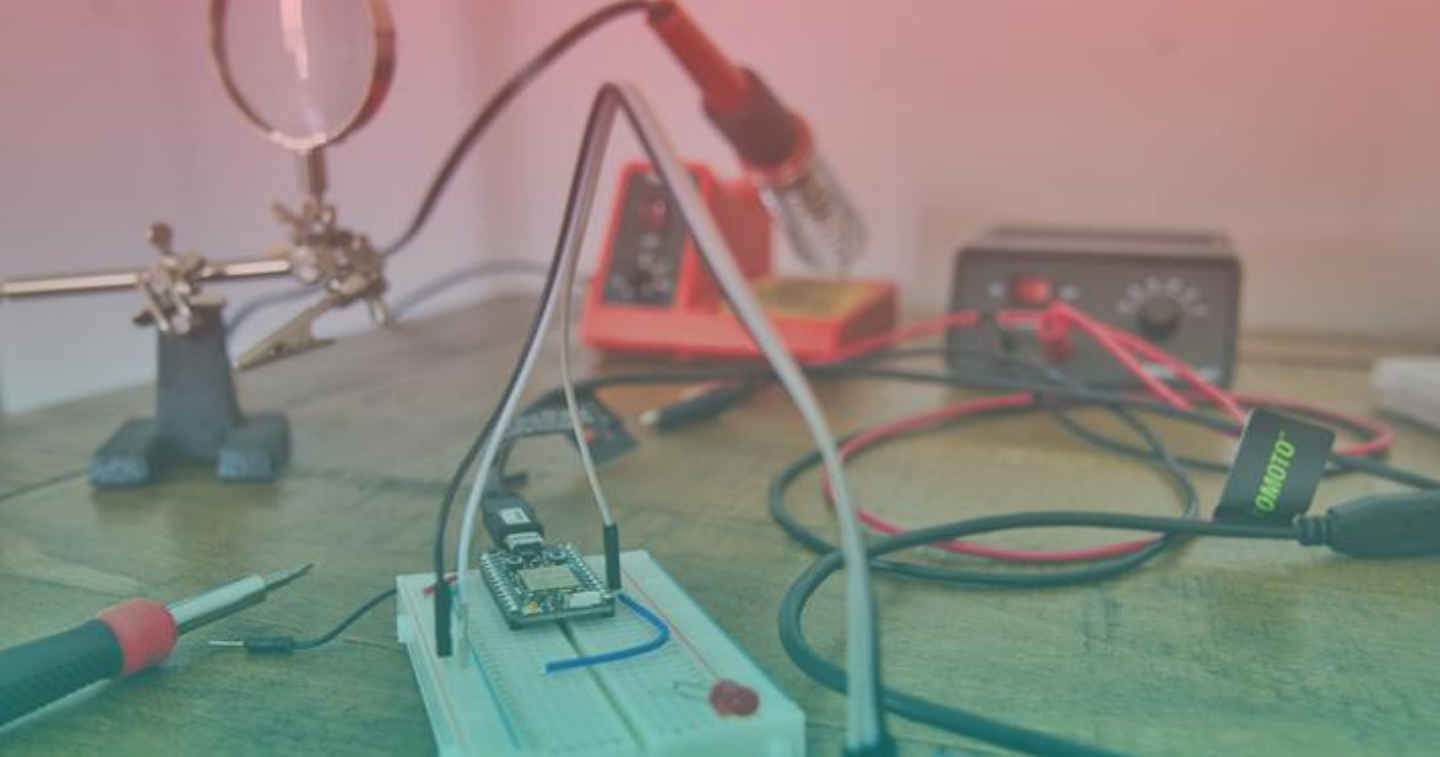


Figura 20 – Fixação da tampa inferior do Robô Ludos.

Depois da montagem, é hora de colocar a mão na massa e fazer o nosso Robô Ludos funcionar.

No próximo tópico iremos realizar uma atividade para trabalhar com todos os recursos do Robô Ludos.



Hora da mão na massa

Há vários tipos de experiências interessantes que você pode realizar com o Robô Ludos. Entre elas, vamos desenvolver uma régua ultrassônica com ele.

Régua Ultrassônica

Desenvolva uma régua ultrassônica com o robô. Pressione o botão e o robô deverá calcular a distância com o sensor ultrassônico e apresentar o seu valor no display LCD.

Se o valor da distância for maior ou igual a 30 cm, os LEDs devem ser acionados. Caso contrário, os LEDs devem permanecer desligados.

A partir do problema apresentado na atividade acima, vamos iniciar desenvolvendo o circuito do projeto.

Utilizaremos o software online TinkerCAD para simulação e montagem com todos os componentes eletrônicos.

A seguir, apresentaremos a estrutura completa do circuito para a solução do problema.

A primeira coisa que você deve fazer é montar o circuito da figura abaixo.

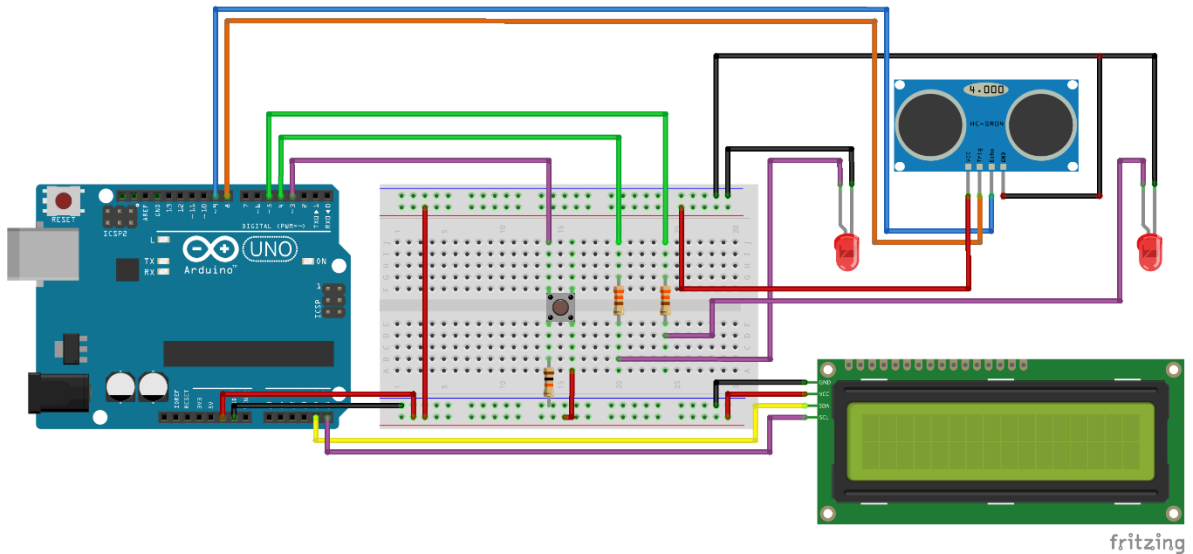


Figura 21 – Esquema Eletrônico do circuito interno do Robô Ludos com o Arduino UNO.

Observe a estrutura acima e veja a conexão de todos os elementos que formam o circuito do robô. Para facilitar a montagem, sugerimos que você utilize o LCD 16x2 com o módulo I2C, pois ele utilizará apenas 2 fios de dados e 2 fios de alimentação.

Agora, veja a lógica completa para solucionar o problema proposto.

Régua Ultrassônica

```
#include <LiquidCrystal_I2C.h> //Biblioteca I2C do
LCD 16x2
#include <Wire.h> //Biblioteca de Comunicacao I2C

LiquidCrystal_I2C lcd(0x27,16,2); // Configurando o
endereço do LCD 16x2 para 0x27

#define botao 3
#define echoPin 9
#define trigPin 8

long tempo = 0;
int distancia = 0;
bool BotaoCabeca = 0, estado = 0;
```

Continuação...

```
void setup()
{
    pinMode(9, INPUT);
    pinMode(8, OUTPUT);

    Serial.begin(9600);

    Wire.begin(); //Inicializacao da Comunicacao I2C
    lcd.init(); //Inicializacao do
LCD
    lcd.backlight();

    lcd.setCursor(2,0);
    lcd.print("Oi, eu sou o");
    lcd.setCursor(3,1);
    lcd.print("Robo Ludos!");
    delay(2000);
}

void loop()
{
    BotaoCabeca = digitalRead(botao);

    if(BotaoCabeca == 1 && estado == 0)
    {

        lcd.clear();
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);

        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);

        tempo = pulseIn(echoPin, HIGH);

        distancia = tempo * (0.034/2);
```

Continuação...

```
        if(distancia <= 30)
        {
            digitalWrite(4, LOW);
            digitalWrite(5, LOW);
        }

        if(distancia > 30)
        {
            digitalWrite(4, HIGH);
            digitalWrite(5, HIGH);

            lcd.print("");
        }

        lcd.setCursor(0,0);
        lcd.print("Distancia:");
        lcd.setCursor(0,1);
        lcd.print(distancia);
        lcd.setCursor(3,1);
        lcd.print("cm");

        estado = 1;
    }

    if(BotaoCabeca == 0 && estado == 1)
    {
        estado = 0;
    }
}
```

A seguir, explicaremos a lógica completa do código desenvolvido. Além disso, mostraremos, em detalhes, o funcionamento de cada elemento utilizado no projeto.

Na primeira parte do código, incluímos a biblioteca do LCD, a biblioteca de I2C e fizemos a sua configuração para um LCD 16x2 e no endereço 0x27.

Essa configuração foi necessária, pois estamos utilizando um LCD com módulo de comunicação I2C. Esse módulo I2C permite que você faça o controle com 4 fios: 2 fios de alimentação e 2 fios de dados. Por fim, definimos nomes para os pinos do sensor ultrassônico e declaramos as variáveis.

O módulo I2C é apresentado abaixo.

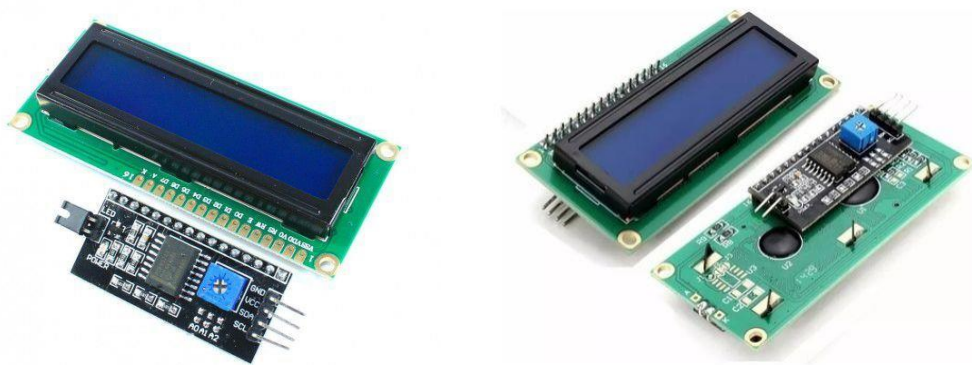


Figura 22 – Display LCD com o Módulo I2C.

A porção de código para a inclusão da biblioteca e configuração do endereço I2C do LCD é apresentada abaixo.

Inclusão da biblioteca e configuração I2C do LCD

```
#include <LiquidCrystal_I2C.h> //Biblioteca I2C do LCD 16x2
#include <Wire.h> //Biblioteca de Comunicacao I2C

LiquidCrystal_I2C lcd(0x27,16,2); //Configurando o endereco do LCD 16x2 para 0x27
```

Em seguida, fizemos a definição de nomes para os números utilizados para manipulação dos pinos digitais e declaramos as variáveis de programa.

A definição de nomes para os números facilita o processo de programação. Isso nos permite substituir o número por nomes quando vamos configurar e acionar os pinos digitais do nosso Arduino.

As variáveis declaradas serão utilizadas para armazenar o estado do botão, o tempo de leitura do sinal do sensor ultrassônico e a distância calculada entre o robô e o objeto.

A porção de código é apresentada abaixo.

Definição de nomes e declaração das variáveis

```
#define botao 3
#define echoPin 9
#define trigPin 8

long tempo = 0;
int distancia = 0;
bool BotaoCabeca = 0, estado = 0;
```

Em seguida, vamos apresentar a função void setup().

Função void setup()

```
void setup()
{
  pinMode(botao, INPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);

  Wire.begin(); //Inicializacao da Comunicacao I2C
  lcd.init(); //Inicializacao do
LCD
  lcd.backlight();

  lcd.setCursor(2,0);
  lcd.print("Oi, eu sou o");
  lcd.setCursor(3,1);
  lcd.print("Robo Ludos!");
  delay(2000);
}
```

Inicialmente, fizemos a configuração dos pinos 3 (botao), 8 (echoPin) e 9 (trigPin) como entradas e saídas digitais.

Em seguida, foi realizada a inicialização da comunicação I2C do display de LCD 16x2.

Posteriormente, fazemos a ativação do backlight (luz de fundo) do LCD e apresentamos a mensagem "Oi, eu sou o Robo Ludos" na tela do LCD.

Após isso, o sistema entra na função void loop e aguarda o momento em que o usuário pressiona o botão.

A função void loop é apresentada a seguir.

Função void loop()

```
void loop()
{
    BotaoCabeca = digitalRead(botao);

    if(BotaoCabeca == 1 && estado == 0)
    {

        lcd.clear();
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);

        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);

        tempo = pulseIn(echoPin, HIGH);

        distancia = tempo * (0.034/2);

        if(distancia <= 30)
        {
            digitalWrite(4, LOW);
            digitalWrite(5, LOW);
        }

        if(distancia > 30)
        {
            digitalWrite(4, HIGH);
            digitalWrite(5, HIGH);
        }
    }
}
```

Continuação da função void loop()

```
        lcd.setCursor(0,0);  
        lcd.print("Distancia:");  
        lcd.setCursor(0,1);  
        lcd.print(distancia);  
        lcd.setCursor(3,1);  
        lcd.print("cm");  
  
        estado = 1;  
    }  
  
    if(BotaoCabeca == 0 && estado == 1)  
    {  
        estado = 0;  
    }  
  
}
```

Inicialmente, fazemos a leitura do pino digital do botão e armazenamos o valor lido na variável BotaoCabeca. Em seguida, verificamos se o valor lido é igual a 1 e se a variável estado é igual a 0.

Leitura e análise do estado do botão

```
BotaoCabeca = digitalRead(botao);  
  
    if(BotaoCabeca == 1 && estado == 0)
```

A variável estado é utilizada para armazenar o estado anterior do botão, ou seja, podemos saber se o botão estava ou não pressionado anteriormente.

Essa condição acima verifica se o botão está acionado atualmente e se anteriormente ele estava desacionado (estado == 0).

Caso essa condição seja verdadeira, o código da condição será executado.

O primeiro comando a ser realizado é a limpeza da tela do LCD, que é feita através do comando `lcd.clear()`. Em seguida, é feito o disparo do sensor ultrassônico para realizarmos a leitura da distância.

A porção de código que realiza o disparo do sensor ultrassônico é apresentada a seguir.

Disparo do sensor

```
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);
```

Após o disparo é feita a leitura do tempo em que o pino echo passa em nível lógico alto. Esse tempo é calculado através da função pulseIn. Essa função retorna o valor do tempo em milissegundos.

A porção de código é apresentada a seguir.

Leitura do tempo de nível alto do pino echo

```
tempo = pulseIn(echoPin, HIGH);
```

O tempo retornado pela função pulseIn é armazenado na **variável tempo** e será utilizado para calcular a distância entre o Robô Ludos e o objeto.

A distância é calculada através da equação abaixo.

Equação para cálculo da distância

```
distancia = (tempo * 0.034/2);
```

Essa equação é utilizada para calcular a distância com base na velocidade média (340 m/s) que o som se propaga pelo ar.

Observe que há um tempo para que a onda possa ir até o alvo e volte até o receptor do sensor ultrassônico. Esse tempo de ida e volta não pode ser utilizado para calcular a distância, pois só precisamos do tempo de ida ou do tempo de volta.

Por esse motivo, nós multiplicamos o tempo pela velocidade de 0.034 e dividimos por 2. Dessa forma, teremos o valor da distância entre o Robô Ludos e o seu alvo.

Após esse cálculo verificamos se o valor da distância é maior ou menor que 30 cm. As condições são apresentadas a seguir.

Verificação do valor da distância calculada

```
if(distancia <= 30)
{
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
}

if(distancia > 30)
{
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);
}
```

Se o valor da distância for menor ou igual a 30 cm, os LEDs devem ser desligados. Caso contrário, os LEDs devem ser acionados.

Após isso, apresentamos o valor da distância no display de LCD e atualizamos o valor da variável estado para 1.

Isso serve para indicar que atualmente o botão está pressionado e evita que ele entre novamente nessa condição. A porção de código é apresentada a seguir.

Apresentação do valor da distância no LCD

```
lcd.setCursor(0,0);
lcd.print("Distancia:");
lcd.setCursor(0,1);
lcd.print(distancia);
lcd.setCursor(3,1);
lcd.print("cm");

estado = 1;
```

Para finalizar, temos a condição que verifica se o botão não está pressionado. A condição é apresentada a seguir.

Condição que verifica se o botão não está pressionado

```
if(BotaoCabeca == 0 && estado == 1)
```

A condição acima verifica se o valor do botão é 0 e se a variável estado é igual a 1. **O que isso significa?**

Ela verifica se o botão não está acionado (`BotaoCabeca == 0`) e se ele estava acionado anteriormente (`estado == 1`). Caso isso seja verdade, a condição será verdadeira e a porção de código abaixo será executada para atribuir 0 à variável estado.

Atualização do estado atual do botão

```
if(BotaoCabeca == 0 && estado == 1)
{
    estado = 0;
}
```

Após isso, o fluxo de código volta para o início do loop e inicia todo o processo novamente.

Viu, como é simples desenvolver esse projeto com o seu Robô Ludos?

Para tornar sua experiência muito mais divertida, vamos oferecer uma série de atividades para você fazer com o seu Robô Ludos. Veja a seguir!

Sugestão de Atividades

- Elabore um programa para acionar os dois LEDs quando o botão for pressionado.
- Elabore um programa para acionar cada LED de acordo com a quantidade de vezes que o botão foi pressionado. Se o botão foi pressionado 1 vez, ligue um LED. Se ele foi pressionado duas vezes, ligue o segundo LED. Caso ele seja pressionado 3 vezes, desligue todos os LEDs e repita o processo novamente.
- Elabore um programa para fazer a leitura da distância com o sensor ultrassônico e pisque os LEDs quando o robô estiver com uma distância menor que 20 cm. Caso o valor seja maior ou igual a 20 cm, os LEDs deverão ser desligados.
- Elabore um programa para apresentar o valor da distância no LCD, sempre que o botão for pressionado.

- Elabore um programa para apresentar no LCD a quantidade de vezes que o botão é pressionado. Se a contagem for maior ou igual a 7, os LEDs devem ser acionados. Entretanto, caso o valor da contagem seja maior ou igual a 15, os LEDs devem ser desligados e a contagem deve ser zerada.
- Utilize outros módulos combinamos com os sensores e elementos do robô Ludos. Utilize, por exemplo, um sensor de temperatura e apresente a informação na tela do Robô Ludos.

Dê asas à sua imaginação e pense em projetos criativos com o robô Ludos. Utilize todo o seu espaço interno, conecte outros elementos ao seu robô e divirta-se com ele!



Documentação técnica do Robô Ludos

Essa é uma das seções mais importantes e você precisa fazer a leitura completa de todos os desenhos. Eles irão te ajudar a entender toda a estrutura de cada peça que compõe o Robô Ludos.

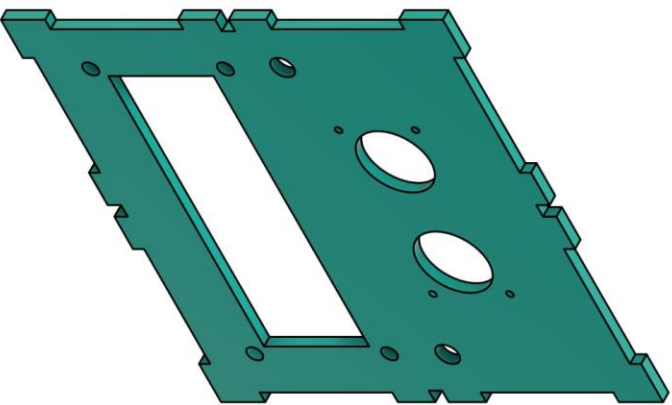
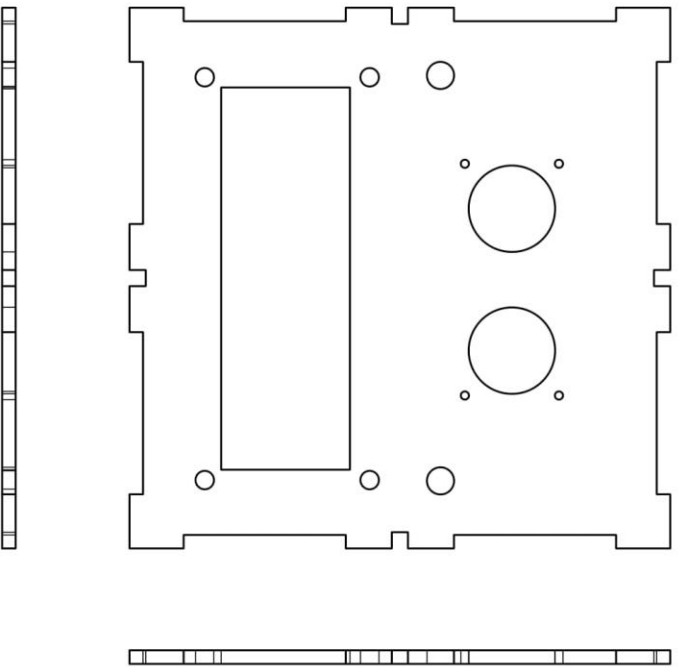
Nas próximas páginas você encontrará o desenho de cada peça e, também, dos parafusos e porcas de fixação das partes do Ludos.

Além disso, você deve ficar atento aos detalhes técnicos para corte das peças à laser. **Você deve utilizar um material de MDF em espessura de 2,5 mm.** Fique atento em relação a isso, pois caso você escolha um outro tamanho, irão ocorrer erros na montagem do robô.

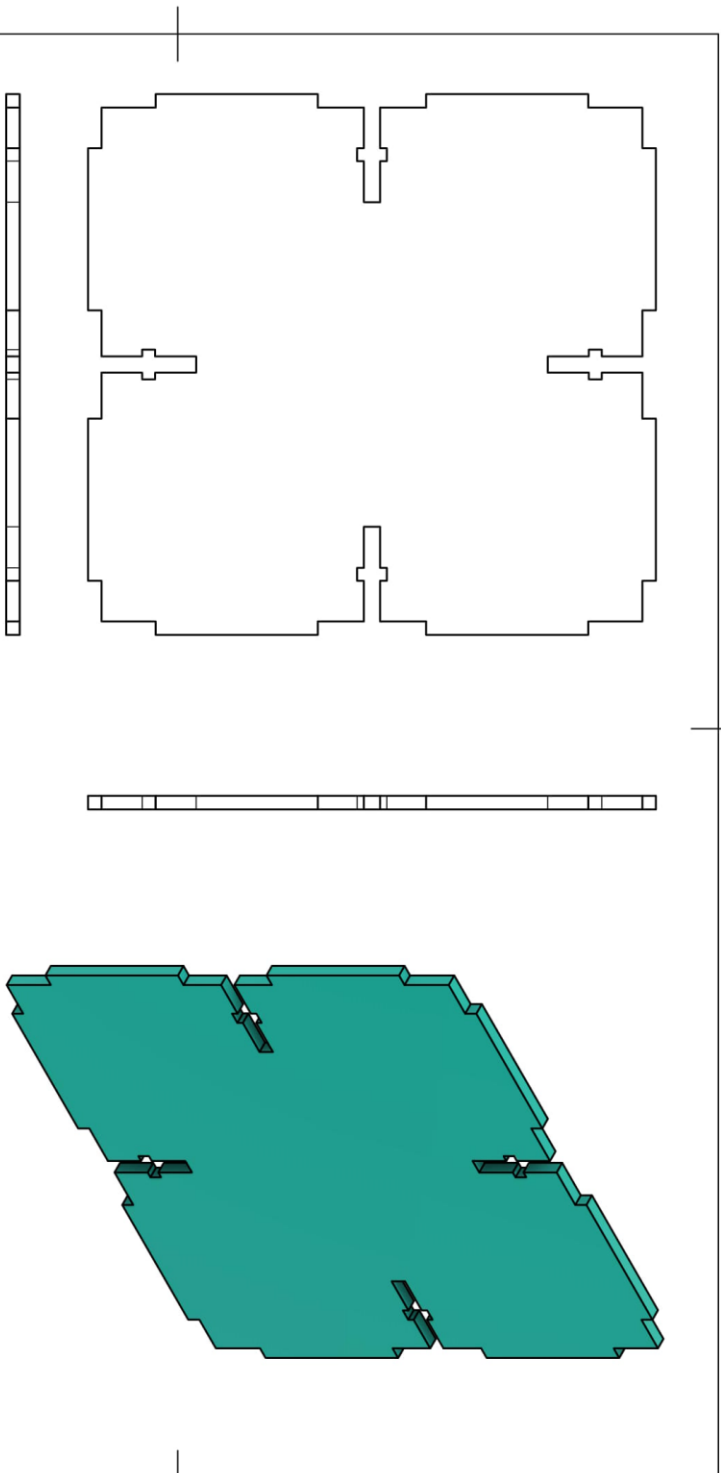
Após o corte à laser ou impressão 3D, adquira seus parafusos e porcas para fixação das partes do seu robô Ludos. Todas as dimensões desses elementos estão representados na documentação técnica.

Para facilitar a compra, envie os arquivos para o vendedor e ele saberá selecionar todos esses parafusos no ato da compra.

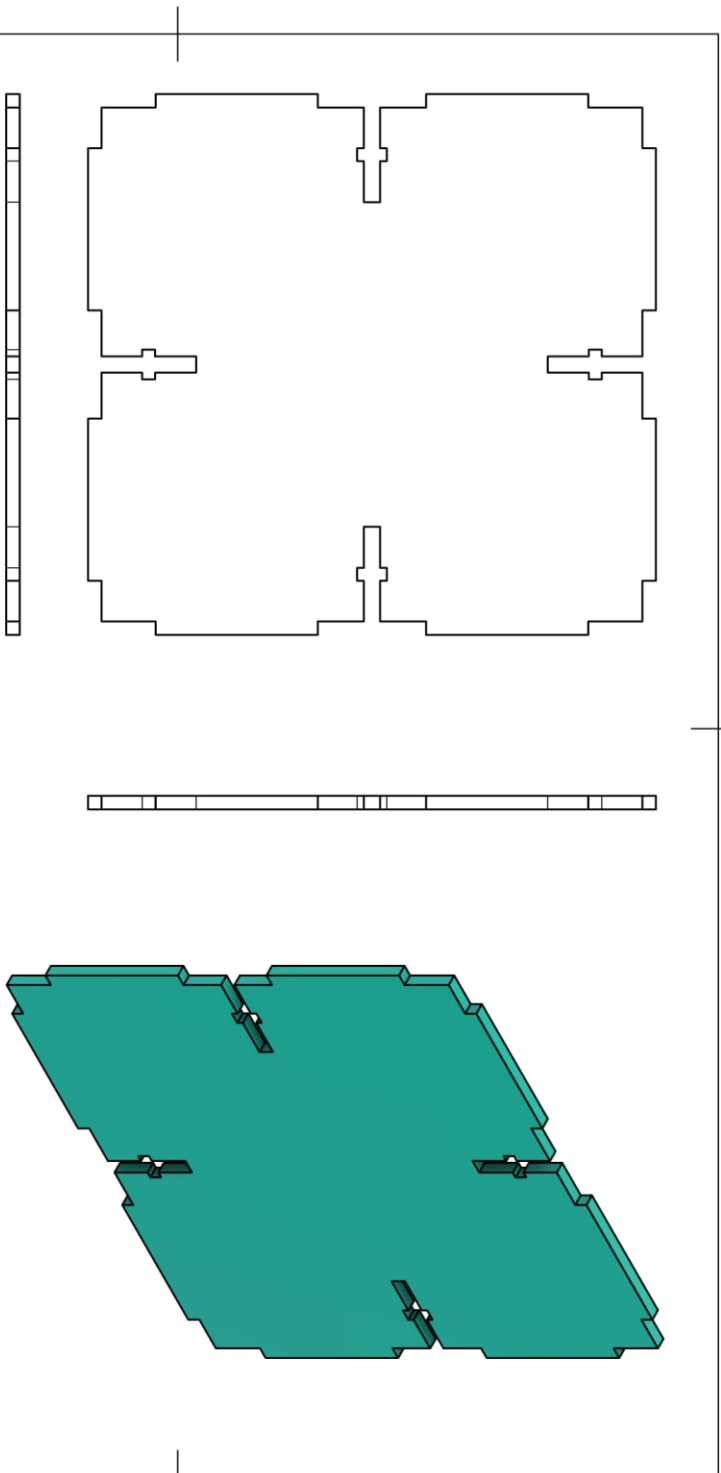
A seguir, disponibilizamos toda a estrutura técnica do Robô Ludos.



Dept.		Technical reference		Created by		Approved by	
				Diego Moreira 05/09/2020		Escola Robô Lúdico	
		Document type		Document status			
		Title		DWG No.			
		Frente Robô					
Rev.		Date of issue		Sheet			
1.0				1/1			

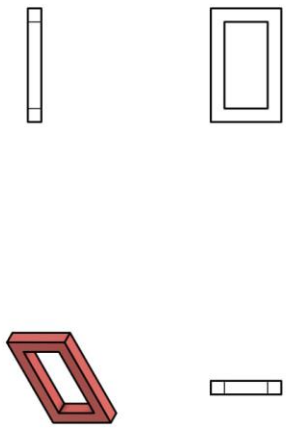


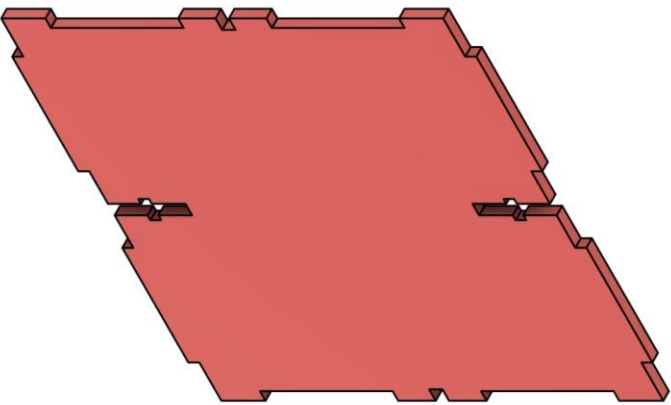
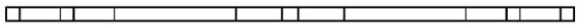
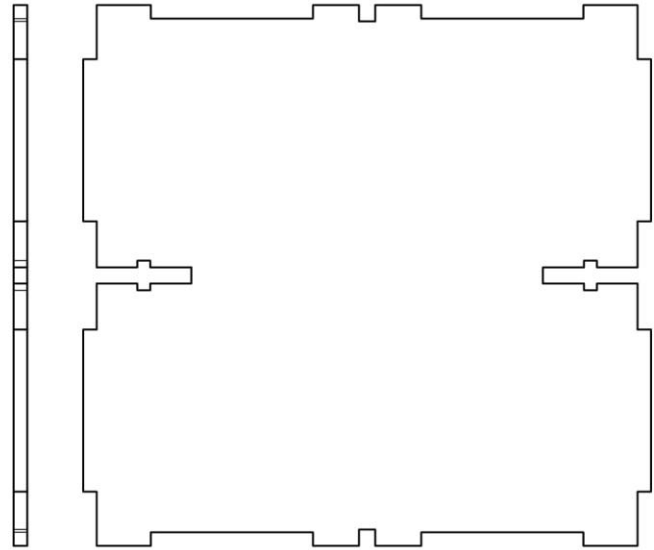
Dept.	Technical reference	Created by	Approved by		
		Diego Moreira	05/09/2020	Escola Robô Lúdico	
		Document type	Document status		
		Title	DWG No.		
		Lateral Direita			
		Rev.	Date of issue	Sheet	
		1.0		1/1	



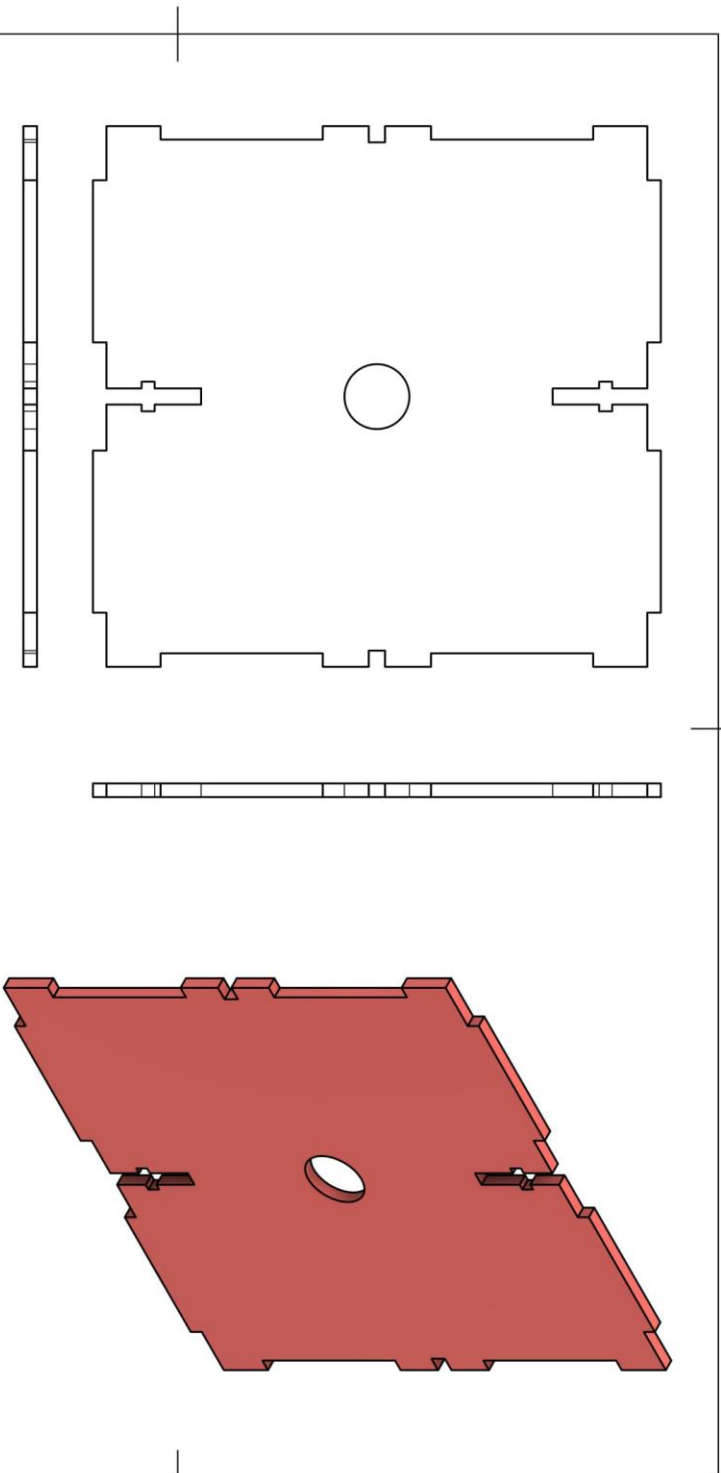
Dept.	Technical reference	Created by	Approved by
		Diego Moreira 05/09/2020	Escola Robô Lúdico
Document type		Document status	
Title		DWG No.	
Lateral Esquerda			
Rev.	Date of issue	Sheet	
1.0		1/1	

Dept.		Technical reference		Created by		Approved by	
				Diego Moreira 05/09/2020		Escola Robô Lúdico	
		Document type		Title		DWG No.	
				Relevo Furo			
Rev.		Date of issue		Sheet			
1.0				1/1			

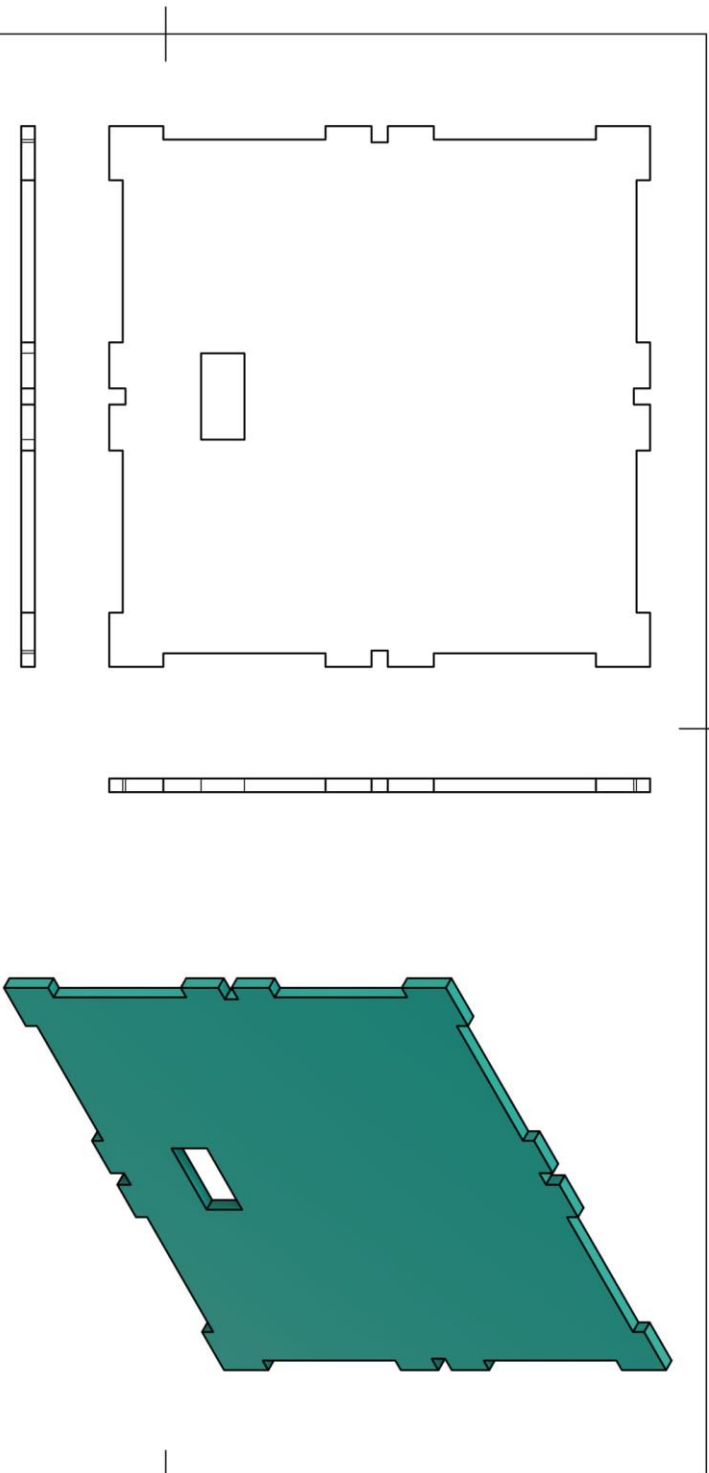




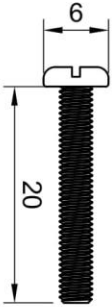
Dept.	Technical reference	Created by	Approved by
		Diego Moreira 05/09/2020	Escola Robô Lúdico
		Document type	Document status
		Tampa Inferior	
		DWG No.	
		Rev.	Date of issue
		1.0	
		Sheet	
		1/1	



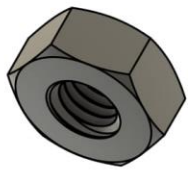
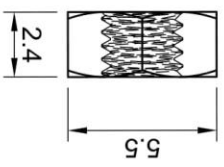
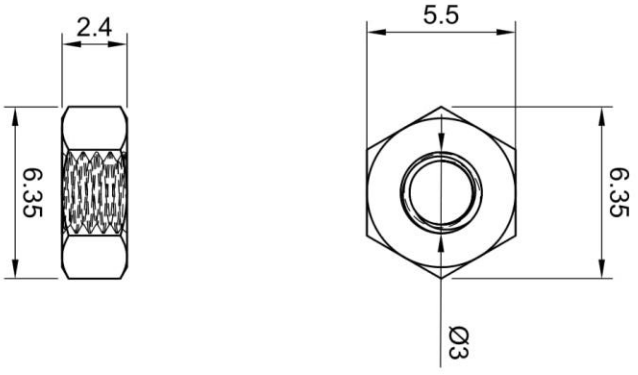
Dept.		Technical reference		Created by		Approved by	
				Diego Moreira 05/09/2020		Escola Robô Lúdico	
		Document type		Document status			
		Title		DWG No.			
		Tampa Superior					
Rev.	Date of issue	Rev.	Date of issue	Rev.	Date of issue	Rev.	Date of issue
1.0		1.0		1.0		1.0	
				Sheet			
				1/1			



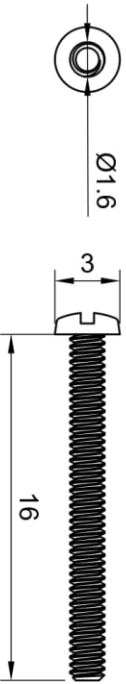
Dept.		Technical reference		Created by		Approved by	
				Diego Moreira 05/09/2020			
		Document type		Document status			
		Title		DWG No.			
		Tampa Traseira					
Rev.	Date of issue	Sheet					
1.0		1/1					



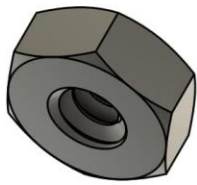
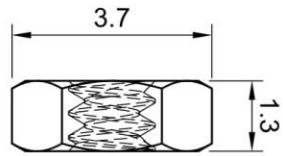
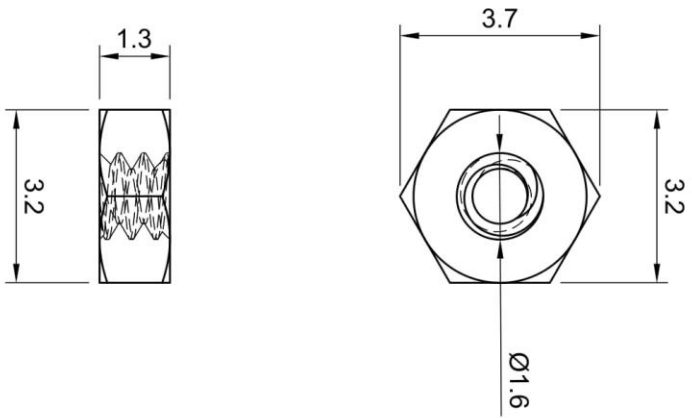
Dept.	Technical reference	Created by	Approved by
		Diego Moreira 05/09/2020	Escola Robô Lúdico
		Document type	Document status
		Parafuso M3	
		Title	DWG No.
Rev.	Date of issue	Sheet	
1.0		1/1	



Dept.		Technical reference		Created by		Approved by	
				Diego Moreira 05/09/2020		Escola Robô Lúdico	
		Document type		Document status			
		Title		DWG No.			
		Porca M3					
Rev.		Date of issue		Sheet			
1.0				1/1			



Dept.	Technical reference	Created by	Approved by
		Diego Moreira 05/09/2020	Escola Robô Lúdico
	Document type		Document status
	Title	DWG No.	
	Parafuso M1.6		
Rev.	Date of issue		Sheet
1.0			1/1



Dept.	Technical reference	Created by	Approved by
		Diego Moreira 05/09/2020	Escola Robô Lúdico
		Document type	Document status
		Title	DWG No.
		Porca M1.6	
Rev.	Date of issue	Sheet	
1.0		1/1	



Suporte técnico e Grupo do WhatsApp do Robô Ludos

Primeiramente, eu agradeço seu apoio e confiança ao nosso trabalho. Todo esse material foi desenvolvido com muito carinho e pensando, principalmente, no seu aprendizado. Meu muito obrigado! 😊

Em caso de dúvidas no processo de montagem ou algum problema, entre contato através do WhatsApp abaixo. Clique no link e fale diretamente com o Diego Moreira.

[Fale agora com o Diego Moreira!](#)

Além disso tudo, criamos um grupo para todos os amantes do Robô Ludos. Vocês serão avisados sobre novidades e atualizações de todo o nosso material.

Lembre-se: Você receberá gratuitamente qualquer atualização futura no material do Robô Ludos.

[Participar do Grupo do Robô Ludos](#)

Chegamos ao fim!

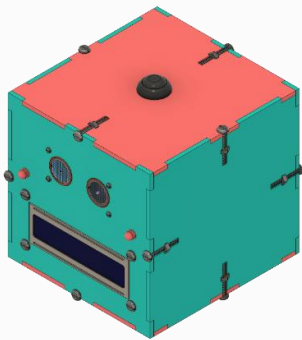
Quanta felicidade eu tenho de você ter chegado até aqui.

Eu imagino que sua experiência foi muito boa ao programar e dar os primeiros passos na robótica com o Robô Ludos.

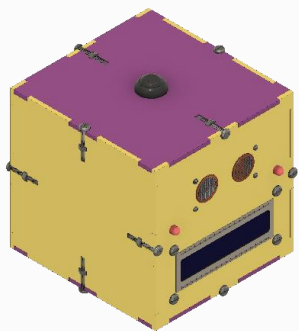
Uma coisa que gostamos: ouvir você!

Entra em contato comigo pelo WhatsApp, nos fala da sua experiência e o que podemos fazer para tornar o Robô Ludos muito mais divertido.

Te esperamos! 😊



Felicidade, diversão, brincadeira, decoração e interação garantida.



Praticidade, simplicidade e facilidade de uso em um único kit.

