

duobit
TECHNOLOGY FOR MAKERS

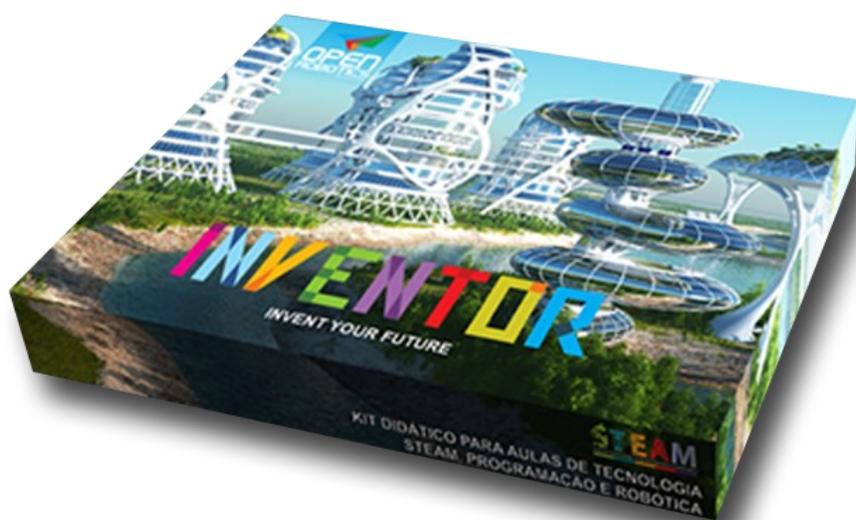


LIVRO DO ESTUDANTE

ROBÓTICA EDUCACIONAL

duobit

TECHNOLOGY FOR MAKERS



Direitos autorais reservados © DUOBIT

Para executar as atividades deste apostila, você vai precisar do kit Inventor, o qual possui dezenas de sensores, motores, peças e ferramentas para montagem de estruturas.

duobit.com.br

Iniciando na robótica

Nas páginas seguintes você vai aprender como fazer a instalação, configuração e primeiros testes da **CPU DARWIN para aulas de robótica educacional**. Qualquer dúvida entre em contato com nossa central de atendimento através do nosso site **duobit.com.br**.

Dispositivos de entrada e saída

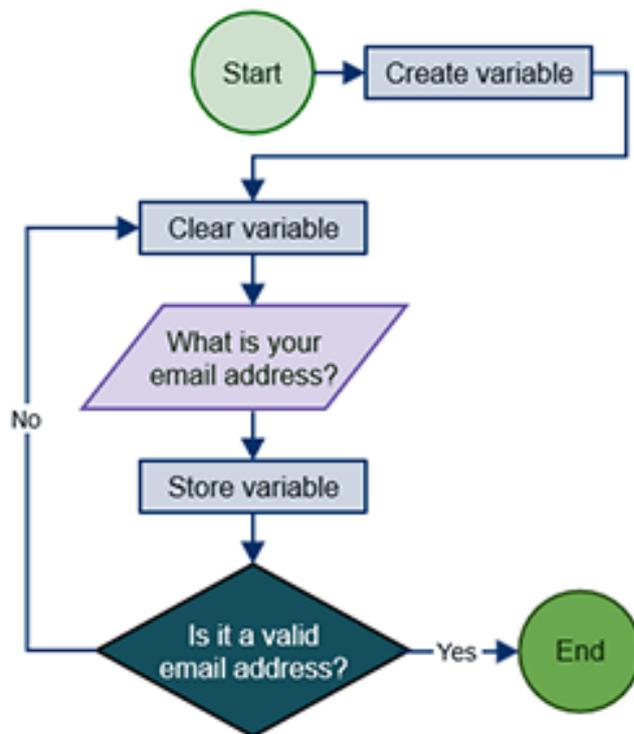
Em cada nova etapa deste tutorial você terá uma visão de algum dos sensores que acompanha o kit, ou seja, você vai aprender a lógica, pela prática, dos dispositivos de entrada e saída, os motores, e todos os outros hardwares que **compõem o seu kit**. É importante você perceber que o conjunto de sensores de entrada e saída, funcionando em conjunto com a parte mecânica, é que vai proporcionar o seu aprendizado. Neste tutorial você vai aprender como os sensores podem ser utilizados, e como programa-los para que façam o que você deseja.

O que é um programa de computador

Um programa de computador, ou software, é uma sequência de instruções que são enviadas para uma CPU (Unidade Central de Processamento). Cada tipo de microprocessador (cérebro) entende um conjunto diferente de instruções, ou seja, o seu próprio "idioma". Também chamamos esse idioma de linguagem de máquina.

As **linguagens de máquina** são, no fundo, as únicas linguagens que os computadores conseguem entender, só que elas são muito difíceis para os seres humanos entenderem. É por isso que nós usamos uma coisa chamada linguagem de programação.

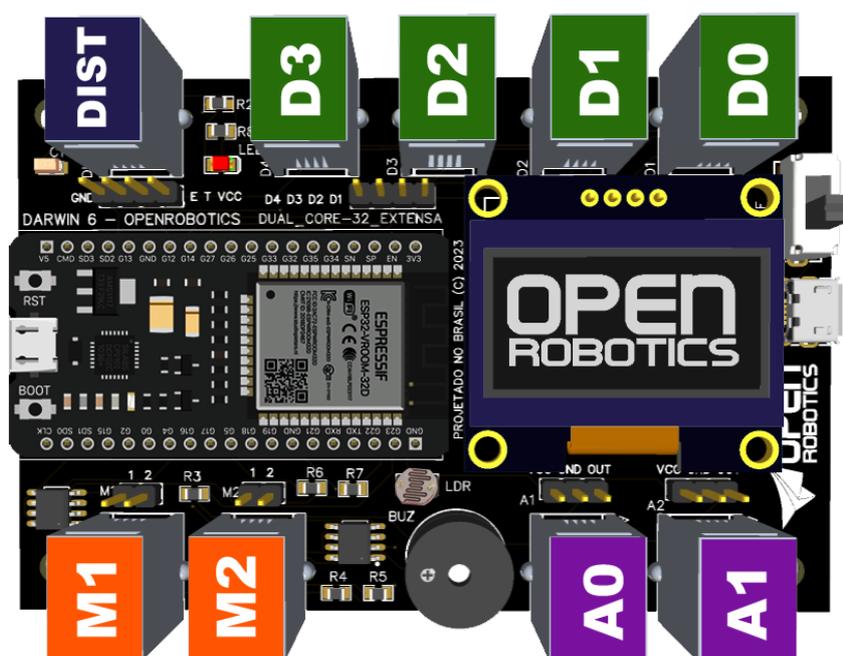
Um algoritmo, ou simplesmente programa, é um jeito de dizer para um computador o que ele deve fazer, de uma forma que nós humanos conseguimos entender facilmente. Os algoritmos normalmente são escritos em linguagens de programação de alto nível. Isso se aplica a praticamente qualquer computador, inclusive ao seu **KIT INVENTOR**.



Conhecendo a CPU DARWIN

Vamos agora conhecer melhor a nossa placa principal denominada **CPU DARWIN**, responsável por realizar todo o controle dos dispositivos que acompanham o seu kit.

O processador que acompanha essa placa chama-se **DARWIN**, o qual pode realizar dezenas de operações lógicas e matemáticas, além de ativar dispositivos de saída e receber informações do ambiente através de sensores. Este processador é chamado de **ESP32**, e possui uma boa capacidade de processamento, contando ainda com Bluetooth e Wifi embutidos.



Nos conectores digitais você vai ligar a **SIRENES**, os **LEDS**, e os **SENSORES DE TOQUE**. O sensor de **distância** por **ultrassom** possui um conector próprio chamado **DIST**. Ao lado dele temos os conectores para os **Motores CC** denominados **M1 E M2**. Um pouco mais a direita da placa temos os conectores de entrada de energia. Abaixo da placa temos os conectores para sensores analógicos tais como sensores de **Luz, som, cor, inclinação**. Nas portas digitais **D0** e **D3** podemos ligar os **servo motores**. Na placa temos ainda um LED, uma Sirene, um sensor de Luz (LDR), display OLED e a chave de liga e desliga.

A placa **DARWIN** trabalha com energia de 5 a 6 Volts conectados via USB, acima do processador, ou na lateral da placa.

A maior parte do tempo você deverá utilizar a energia disponível via USB do seu computador, mas acompanha o seu kit um power bank, o qual pode ser utilizado em situações remotas, onde você não esteja próximo do computador.

Conhecendo os dispositivos

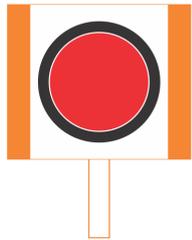
Os sensores compõem uma parte importante de qualquer robô. São eles os responsáveis em permitir que o robô receba dados do ambiente. É possível termos sensores para diversas situações, por exemplo, sensores de cor, de temperatura, de humidade, de distância, de ultrassom, e diversos outros.

Vejam os pelas imagens abaixo quais os sensores você está recebendo no seu INVENTOR. Atenção: a quantidade e o tipo de sensor podem variar de uma versão para outra.



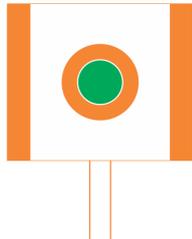
Sensor de Luz

Possui uma abertura frontal que permite a passagem de luz. Ele percebe se o ambiente está Claro ou Escuro. Você pode criar um robô que quando entra em algum local escuro, então ele acende uma lanterna.



Sensor de Toque

Este sensor possui um interruptor chamado push-botton. Ou seja, quando o pressionamos, ele emite um sinal de positivo para o robô.



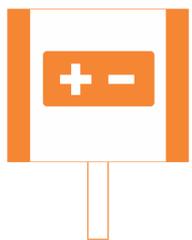
Lâmpada de Led

No seu INVENTOR você recebeu 3 lâmpadas de LED (Vermelho, verde, amarelo). Elas podem ser ligadas para sinalizar alguma situação importante, ou avisar sobre uma tomada de decisão pelo robô,



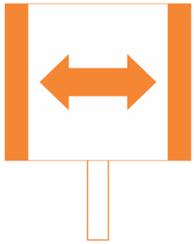
Sensor de som

Este sensor possui 8 furos frontais por onde pode entrar qualquer tipo de ruído. Ele pode ser utilizado para que o robô seja capaz de escutar o ambiente.



Sensor de hall

Este sensor pode sentir a presença de campos magnéticos. Com ele você pode detectar materiais magnéticos como ímãs.



Sensor de inclinação

Este é um sensor muito interessante. Com ele podemos verificar se o robô está na vertical ou na horizontal. Isso pode ser muito útil quando você tiver que saber “como” o robô se encontra em um determinado momento. Pode ser que ele esteja de cabeça para baixo por exemplo.



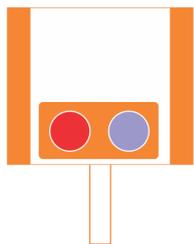
Sirene

Assim como os LEDs a sirene é um dispositivo de saída. Ou seja, uma sirene pode ser utilizada quando você precisar emitir algum sinal sonoro no ambiente.



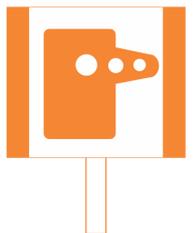
Sensor de distância por ultrassom

Utilizado para medir a distância de objetos, podendo variar de 1cm até 2m de distância. Você pode utilizar este sensor para criar veículos que se movimentam em um ambiente sem tocar nos objetos.



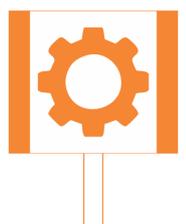
Sensor de infravermelho

Este sensor pode ser utilizado para verificar a presença de obstáculos, ou também para diferenciar cores (preto ou branco) de objetos que estejam a aproximadamente 3cm de distância. Este sensor também pode ser utilizado para criar robôs que se deslocam por ambiente sem tocar nos objetos.



Servo motor

Este motor pode fazer giros precisos entre 0 e 180 graus. Geralmente são utilizados em articulações de robôs e em mecanismos que exigem precisão nos movimentos. Na sua placa você vai encontrar dois motores como este.



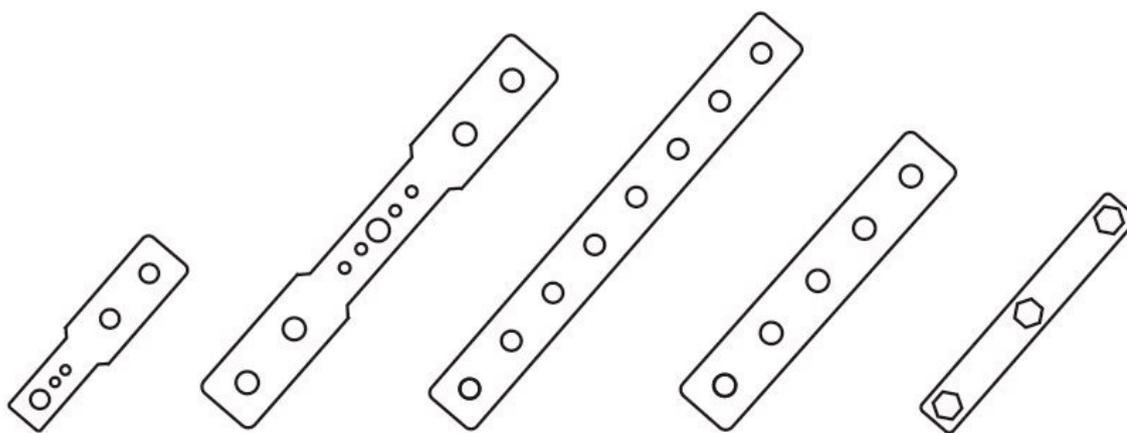
Motor CC

Este motor é o responsável por fazer as rodas de borracha funcionarem. Ele consome muita energia e como dito acima, você deverá ligá-lo diretamente na saída M1 e M2 do seu kit.

Conhecendo os conectores

Seu Kit INVENTOR possui várias peças conectores as quais podem ser utilizadas para dezenas de montagens diferentes. Essas peças foram projetadas para facilitar o encaixe uma nas outras e também nos sensores. Eles são feitos de um material conhecido como **polimetilmetacrilato (PMMA)** um material termoplástico rígido, transparente, colorido ou incolor; também pode ser considerado um dos polímeros (fibra sintética) mais modernos e com maior qualidade do mercado, por sua facilidade de adquirir formas, por sua leveza e alta resistência.

No kit INVENTOR ele possui várias formas e foram criados utilizando ferramenta de corte a laser, portanto, todas as peças possuem um plástico de proteção durante o corte. Você pode manter esse plástico, mas também pode retirá-lo durante o uso. **Você vai perceber que algumas peças estão escuras, aparentemente sujas, mas isso se deve ao calor do raio laser que realizou o corte, portanto, basta remover este plástico e sua peça estará pronta para uso.** Vejamos quais são elas.



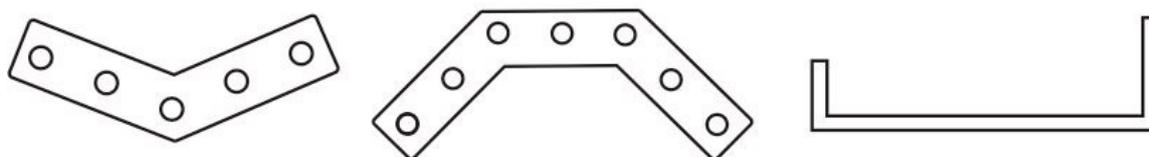
2 Haste F2

1 Haste F4

8 Barras 8F

8 Barras 5F

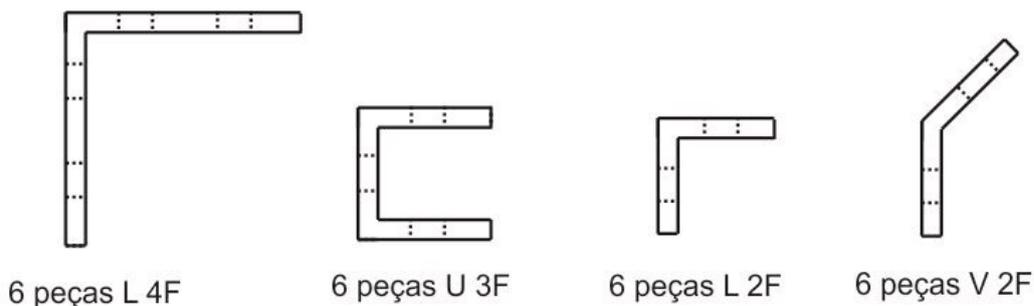
1 Chave boca



6 peças V 5F

6 peças Cone 7F

1 Chassi para veículo



6 peças L 4F

6 peças U 3F

6 peças L 2F

6 peças V 2F

Instalando a placa DARWIN

Chegou a hora de preparar a sua placa para funcionamento. Veremos passo a passo todo o processo de instalação e configuração do seu computador para que você possa programar e enviar os comandos para seu kit INVENTOR.

Passo 1: Download dos programas

Para baixar os programas de instalação entre no site www.duobit.com.br/downloads.

1. Encontre a seção denominada **KIT INVENTOR 6.0**
2. Clique em : **Ambiente de programação Arduino + Biblioteca Darwin.h**

O arquivo que será baixado está compactado. Dentro dele temos o programa instalador (a versão 1.8 do Arduino) e uma pasta com algumas bibliotecas de recursos necessários ao correto funcionamento do seu **INVENTOR**.

Após baixar este arquivo, descompacte em algum lugar apropriado no seu computador. Veja como deverá ficar:

Nome	Data de modificação	Tipo
Darwin	09/06/2024 20:07	Pasta de arquivos
Instalação Arduino 1.8.exe	09/06/2024 20:02	Aplicativo

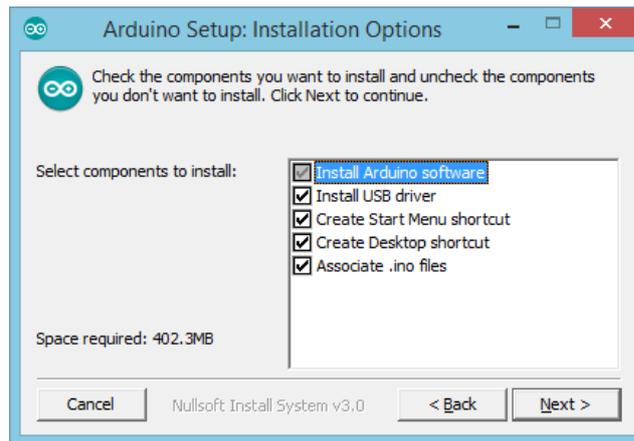
Passo 2: Disparando a instalação

Agora que você já possui uma pasta com o programa de instalação, basta executar a **Intalação Arduino**, clicando 2 vezes sobre ele.

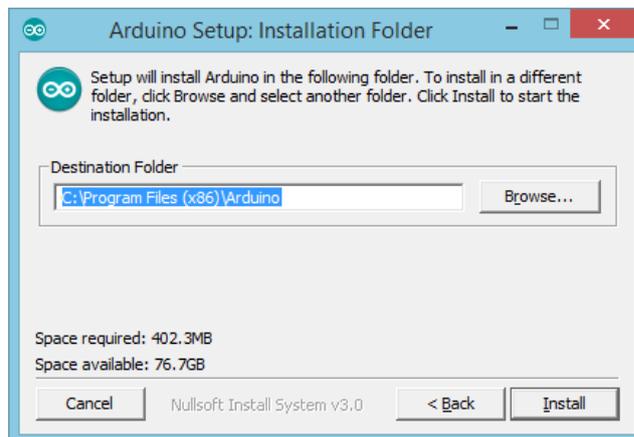
Você terá uma tela como mostrada ao lado, basta clicar na opção **I Agree**.



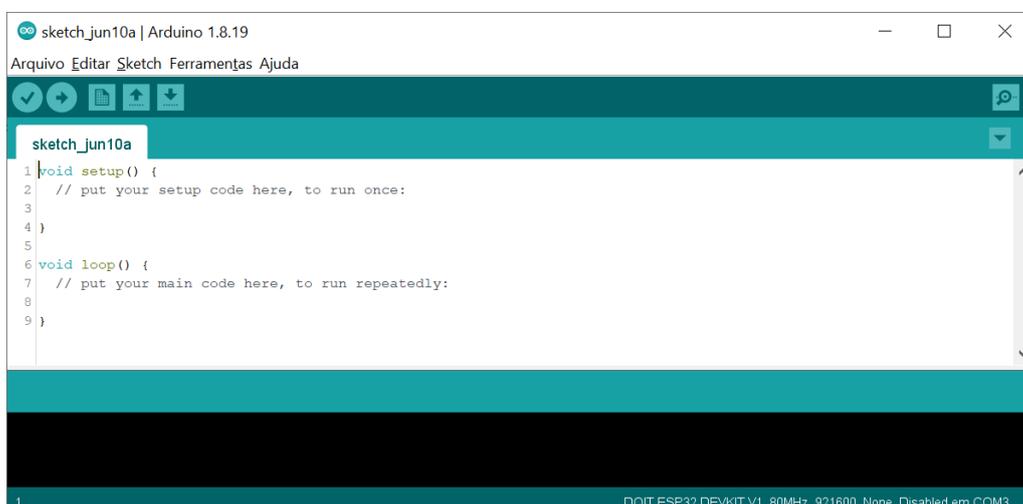
Na tela seguinte você pode clicar em **NEXT** para aceitar todas as opções de instalação.



Na sequencia vai aparecer uma tela com a opção de pasta onde você deseja instalar os programas do seu kit. Você **pode deixar a opção padrão** ou alterar conforme a sua necessidade. Clique no botão **Install** e aguarde o término da instalação.



Ao término da instalação um ícone chamado ARDUINO será colocado em sua área de trabalho. **Clicando duas vezes sobre ele** você já consegue abrir o ambiente de programação.



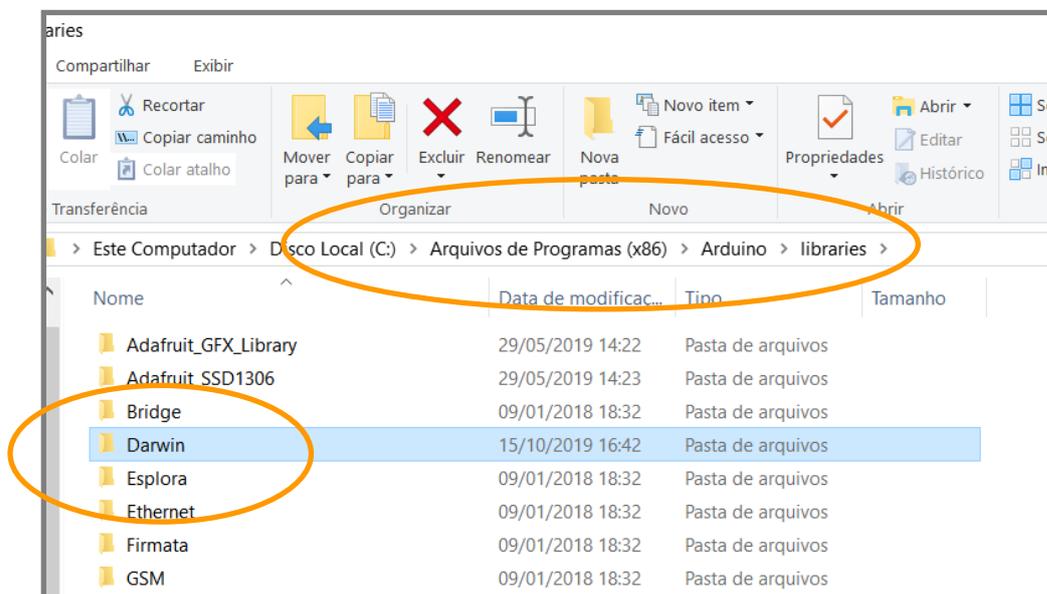
Passo 3: Copiando a biblioteca do Inventor

Agora que a instalação já está pronta, falta pouco para ficar tudo resolvido e você iniciar os seus primeiros testes. No local onde você salvou os seus arquivos tem uma pasta chamada **Darwin**. Ela contém a **biblioteca de comandos** necessários ao correto funcionamento do kit.

Nome	Data de modificação	Tipo
Darwin	09/06/2024 20:07	Pasta de arquivos
Instalação Arduino 1.8.exe	09/06/2024 20:02	Aplicativo

Faça o seguinte:

- Clique com o botão direito na pasta **Darwin** e escolha a opção **Copiar**.
- Vá até a pasta **Libraries** que está no local onde você instalou o Arduino, por exemplo: **C:\Program Files (x86)\Arduino\libraries**
- Cole a pasta **Darwin** dentro da pasta **libraries**, para que fique como abaixo:



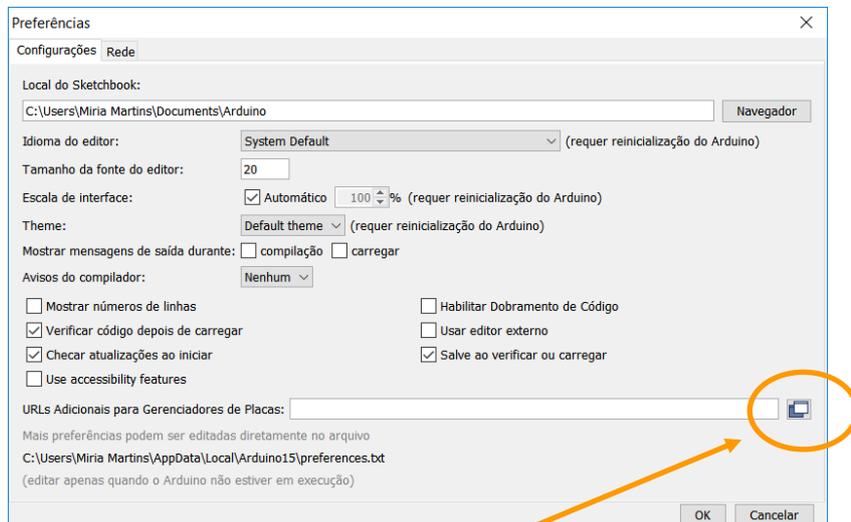
No próximo passo vamos instalar o **driver de comunicação** entre o computador e a placa Darwin.

Passo 4: Instalando o processador ESP32

Neste ponto já estamos com o Arduino instalado e também a biblioteca de comandos para a placa Darwin.

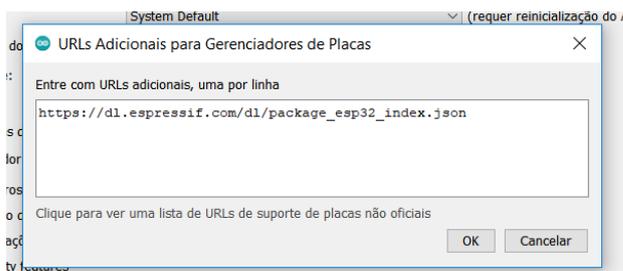
Agora precisamos instalar o processador ESP para que ele apareça na lista de placas disponíveis.

- Dentro do Arduino, clique no menu **Arquivo -> Preferências**. Você vai ver uma janela como essa mostrada abaixo:

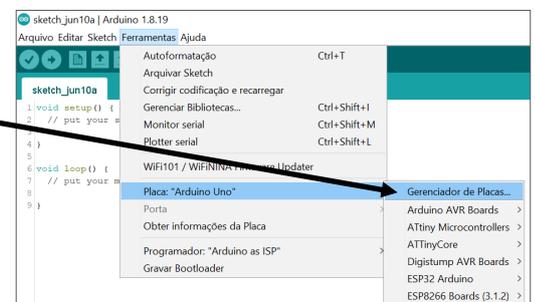


- Clique no botão URLs adicionais
- Digite a seguinte linha na janela que vai se abrir e clique em OK para fechar as janelas que estão abertas:

https://dl.espressif.com/dl/package_esp32_index.json

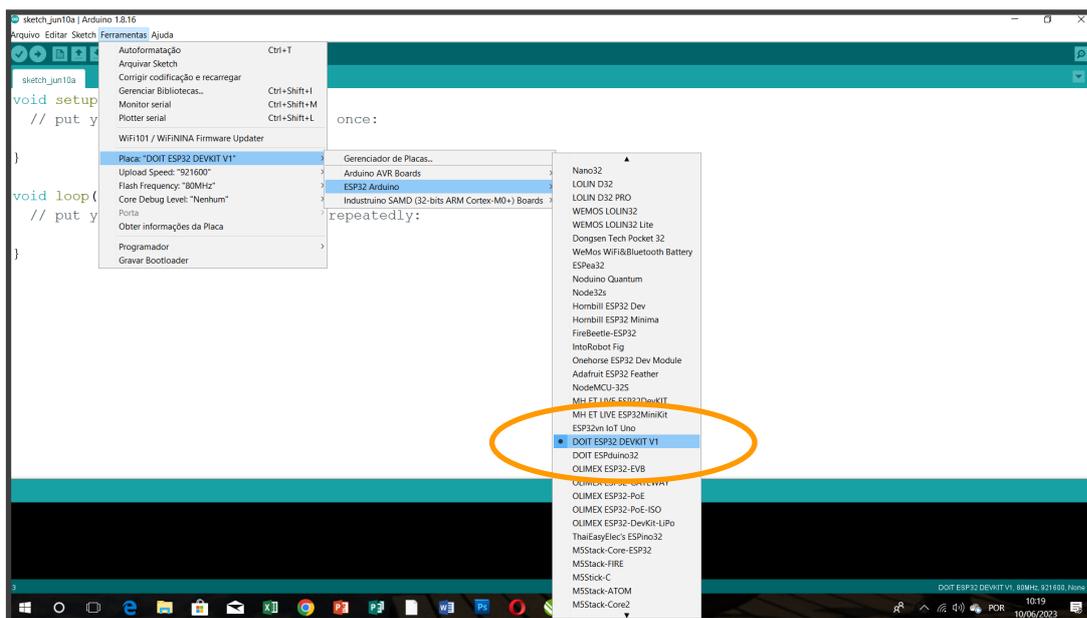


- Agora clique no menu **Ferramentas -> Gerenciar Placas**.
- Na tela que vai aparecer, digite **esp32**.
- Escolha a versão **2.0.11** e clique em **Instalar**



Pronto. Agora você já pode escolher a placa no painel do Arduino.

- Escolha a opção **DOIT ESP32 DEVKIT V1**



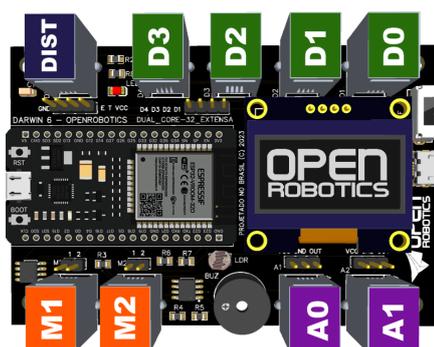
Instalando o driver da placa

Agora que já configuramos o Arduino para poder enviar programas para a placa Darwin, vamos instalar a placa no computador. Para essa instalação, você vai precisar saber se a versão do seu Windows é 7, 8, 10 ou superior.

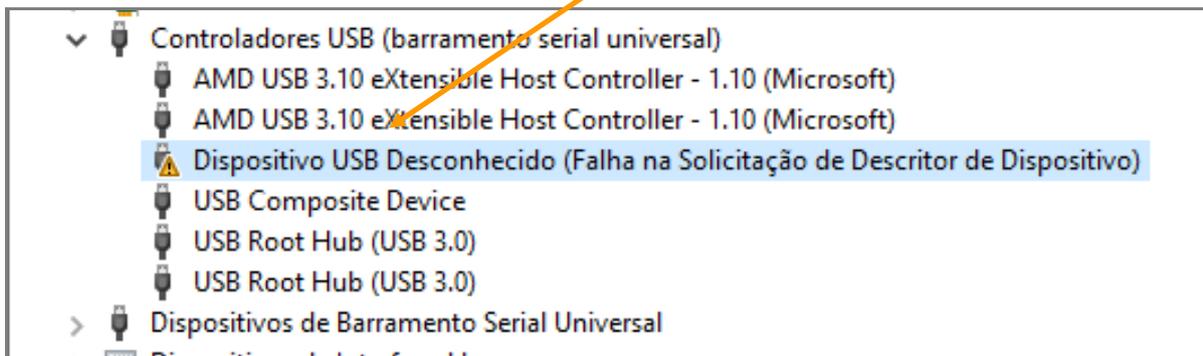
- Acesse o site da **DUOBIT** para baixar o driver no seguinte endereço:
[Duobit.com.br/downloads](https://duobit.com.br/downloads)
- Clique no link **Driver USB da placa Darwin**, faça o download.
- Após o download, descompacte os arquivos em uma pasta temporária.

Nome	Data de modificação	Tipo
Driver USB placa Darwin Windows 7 e 8	10/06/2024 07:59	Pasta de arquivos
Driver USB placa Darwin Windows 10 e 11	10/06/2024 07:59	Pasta de arquivos

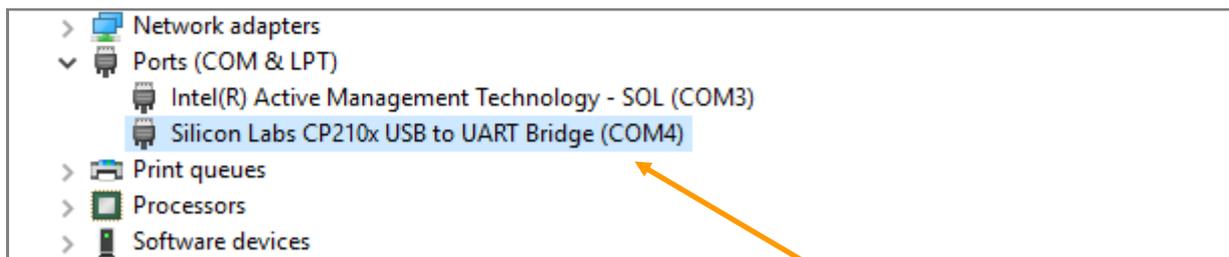
- **Ligue a placa Darwin em seu computador.**



Acesse o **Gerenciador de dispositivos** no seu computador e veja que na seção **Controladores USB** a placa irá aparecer com erro, pois ainda não foi instalada

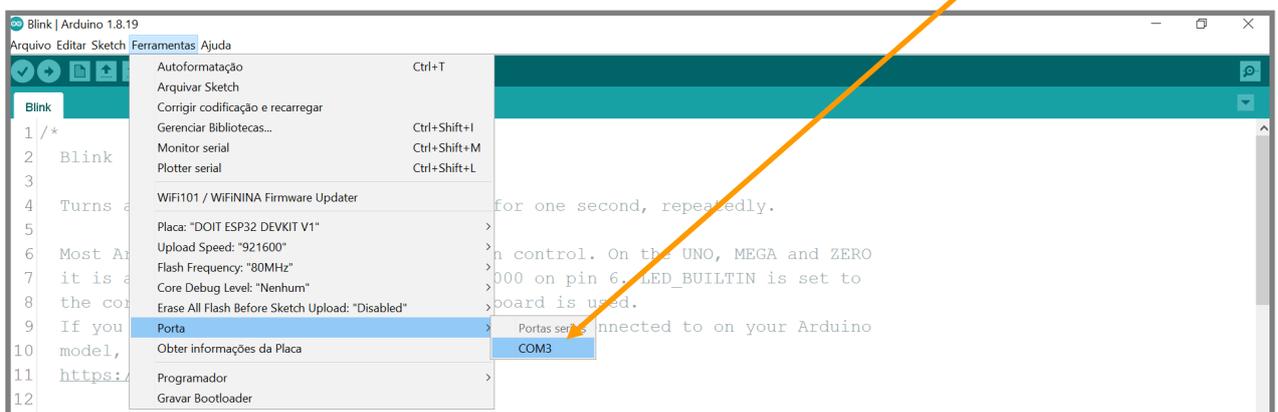


- Clique com o botão direito sobre o Dispositivo USB que ainda não foi reconhecido e escolha a opção **Atualizar Driver**.
- Navegue até a pasta onde você descompactou os drivers e escolha a pasta conforme o seu sistema operacional. **clique em Ok**



Perceba que a instalação (no caso acima) **foi realizada na porta COM4**. Pode ser que no seu caso apareça uma outra porta, e também pode acontecer de mudar essa porta em um outro momento de uso.

- Informe o número da porta dentro do Arduino, acessando o **Menu Ferramentas, Porta**.



Passo 5: Enviando seu primeiro programa

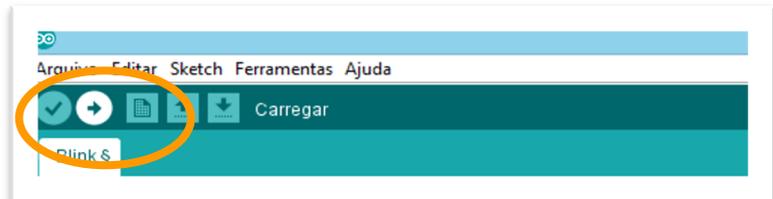
Agora vamos testar o envio de um primeiro programa. Vamos lá:

Digite o código abaixo no seu Arduino e salve-o com o nome de **TESTE1**. Atenção: as letras maiúsculas, minúsculas e **demais símbolos** devem ser digitados rigorosamente como mostrados abaixo. Coloque um LED na porta D3 para experimentar o funcionamento.

```
#include <darwin6.h>

void setup()
{
  LedNaPorta(D3);
}

void loop()
{
  LigarLed(D3);
  EsperarSegundos(1);
  DesligarLed(D3);
  EsperarSegundos(1);
}
```



Após digitar o programa acima, envie para a CPU clicando no botão **Carregar**.

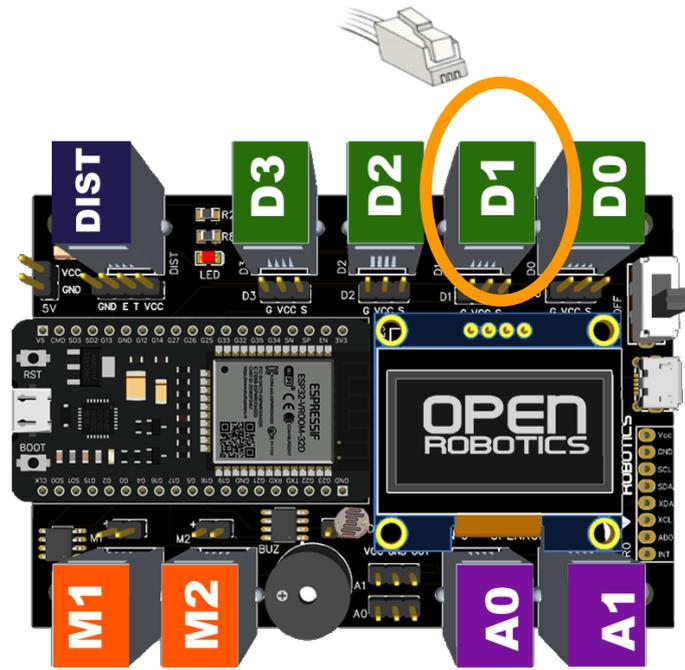
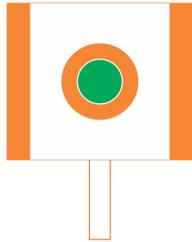
Se tudo der certo o LED que você conectou na porta D3 irá ligar e desligar com intervalo de 1 segundo. Caso algo dê errado, revise seu programa para conferir se tudo foi digitado corretamente. Um sinal de que tudo deu certo é uma mensagem em branco que aparece logo abaixo na tela do **Arduino**.

Nesta mesma tela pode aparecer a mensagem de erro alertando sobre um problema que pode ter acontecido. A primeira vez não é tão fácil, mas com o tempo você vai se acostumar a programar e enviar para a CPU.

Acionando lâmpadas de LED

As lâmpadas podem existir com diversos formatos e em **diversos tipos e funcionamento**. Uma das mais utilizadas hoje tanto pela **economia** quanto pela sua utilidade são as **lâmpadas de LED**. Vejamos como podemos fazer uma delas funcionar.

Fazendo um LED piscar



Programação:

```
#include <darwin6.h>

void setup()
{
  //Informando que vamos ter um LED na porta Digital D1
  LedNaPorta(D1);
}

void loop()
{
  LigarLed(D1);
  EsperarSegundos(1);
  DesligarLed(D1);
  EsperarSegundos(1);
}
```

Desafios desta aula:

1. Coloque duas lâmpadas e faça elas piscarem alternadamente.
2. Faça com que a primeira lâmpada pisque duas vezes e a outra, de forma alternada, pisque uma única vez.

O que é um robô?

Um robô é uma máquina programável que é capaz de realizar tarefas de maneira autônoma ou semiautônoma. Essas máquinas podem variar significativamente em complexidade e função, desde simples dispositivos que executam tarefas repetitivas até sistemas avançados que possuem inteligência artificial e são capazes de aprender e adaptar-se a diferentes situações.

Os robôs podem ser projetados para diversas finalidades, tais como:

Manufatura Industrial: Robôs industriais são frequentemente utilizados em linhas de produção para realizar tarefas repetitivas, como montagem de produtos, soldagem, pintura, entre outros.

Exploração Espacial: Robôs são enviados para explorar outros planetas, luas e ambientes extraterrestres, realizando tarefas como coleta de dados, análise de solo, etc.

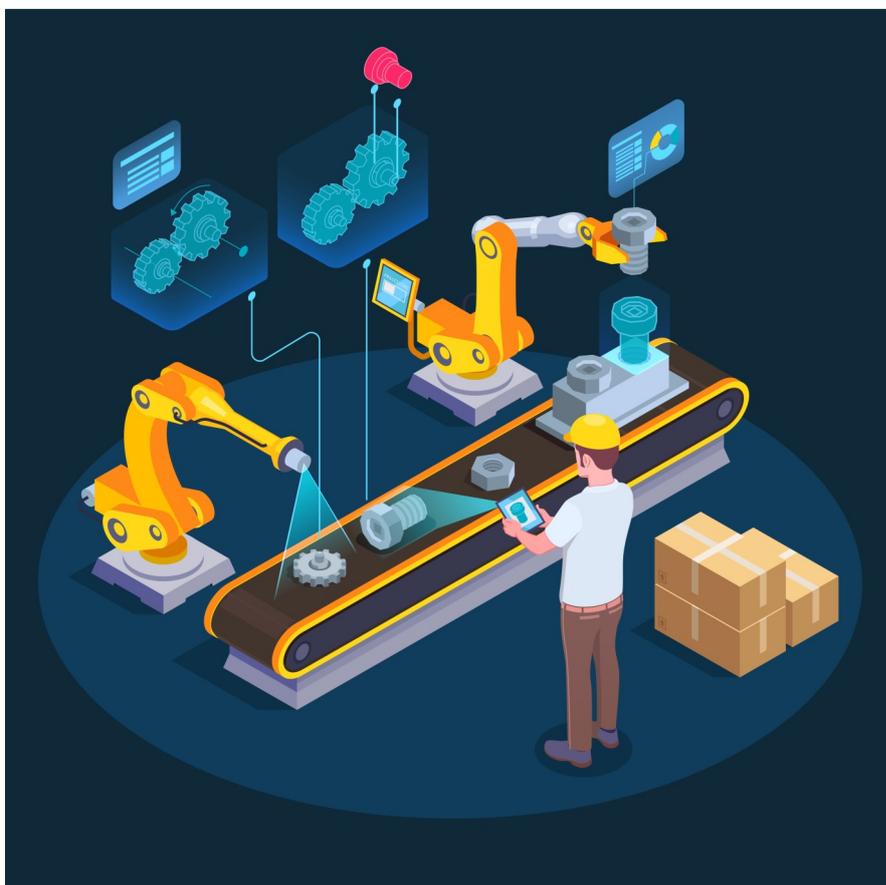
Assistência Médica: Robôs são desenvolvidos para auxiliar em cirurgias, reabilitação, monitoramento de pacientes, entre outras aplicações na área de saúde.

Atendimento ao Cliente: Alguns robôs são programados para interagir com clientes, responder a perguntas frequentes e realizar tarefas simples de atendimento.

Exploração Subaquática: Robôs subaquáticos são usados para explorar o fundo do mar, realizar pesquisas oceanográficas e inspeções submarinas.

Militar: Em aplicações militares, robôs podem ser usados para desarmar bombas, realizar reconhecimento em áreas perigosas, e outras funções.

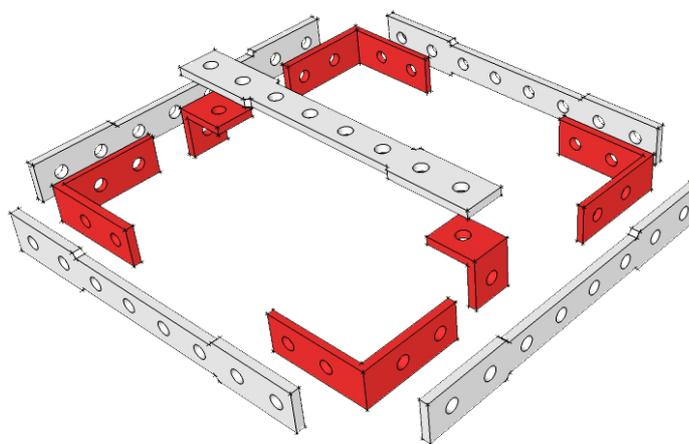
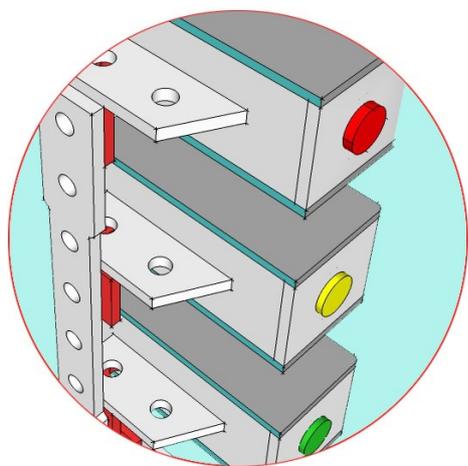
Os robôs podem ser controlados por humanos ou podem operar de forma autônoma, utilizando sensores e algoritmos de controle para tomar decisões com base nas informações do ambiente. Eles desempenham um papel cada vez mais importante em diversas áreas, melhorando a eficiência, segurança e produtividade em muitas atividades.



Montando um semáforo

Vamos agora montar um semáforo (também conhecido popularmente como **sinal, sinaleira, farol ou sinal luminoso**) um instrumento utilizado para controlar o tráfego de veículos e de pedestres nas grandes cidades em quase todo o mundo. Utiliza uma **linguagem simples** e por isso de fácil assimilação. É composto geralmente por três círculos de luzes coloridas.

Geralmente o semáforo é controlado por um computador igual a este que você vai utilizar na sua montagem. O semáforo vai deixar os veículos passarem por um determinado tempo, e ficarem parados por um outro período de tempo. Tudo depende do fluxo desejado, seja de pedestres, seja de veículos.



Você vai precisar dos seguintes materiais: 1 LED vermelho, 1 LED amarelo, 1 LED verde, 7 barras de 8 furos, 5 peças L com 4 furos, 5 peças L com 2 furos, parafusos e porcas.

Para montar o semáforo faça o seguinte:

1. Comece montando a base com **5 barras longas de 8 furos** e também **4 peças L de 4 furos** nos cantos. Em duas laterais coloque **1 peça L de 2 furos**, sobre as quais você vai fixar **1 barra de 8 furos**. No centro, sobre essa barra de 8 furos, acrescente uma peça **L com 4 furos**, e nela fixe **2 barras longas com 8 furos** cada.
2. Utilize as peças em **L de 2 furos** para fixar cada um dos LEDs no poste que sustenta o semáforo.
3. Os 3 LEDs deverão ser ligados na porta **D3, D1 e D2** respectivamente (**vermelho, amarelo, verde**).

Programando o semáforo

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduíno**.

1. Antes de digitar e enviar o programa, ligue **o LED vermelho na porta D3, o LED amarelo na porta D1 e o LED verde na porta D2**.

2. Na área de trabalho do seu computador dê dois cliques no ícone do **Arduíno**.
3. Então crie o programa mostrado abaixo com a ajuda do professor.

Programação :

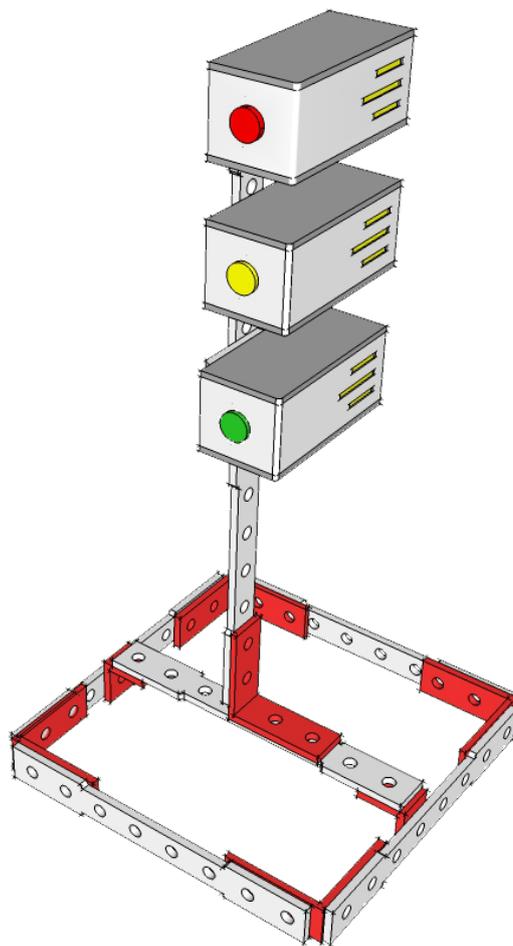
```
#include <darwin6.h>

void setup() {
  //Adicionando os Leds em cada porta
  LedNaPorta(D3);
  LedNaPorta(D1);
  LedNaPorta(D2);
}

void loop()
{
  LigarLed(D3);
  EsperarSegundos(8);
  DesligarLed(D3);

  LigarLed(D2);
  EsperarSegundos(5);
  DesligarLed(D2);

  LigarLed(D1);
  EsperarSegundos(2);
  DesligarLed(D1);
}
```



Entendendo o que o programa faz:

Nosso programa vai fazer com que o LED vermelho fique aceso por **8 segundos**, em seguida ele se apaga. Depois o LED verde vai ficar aceso por **5 segundos**, e se apaga. Ao final o LED amarelo fica aceso por **2 segundos** e também se apaga. Esta rotina vai ficar em execução até que a placa seja desligada, ou seja, vai ficar em **loop** enquanto houver energia. Experimente mudar estes tempos e veja os resultados.

Ampliando seus conhecimentos

Após a realização da **montagem e da programação** da estrutura do semáforo, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Você acha tranquilo o trânsito na cidade onde você mora? Poderia melhorar?

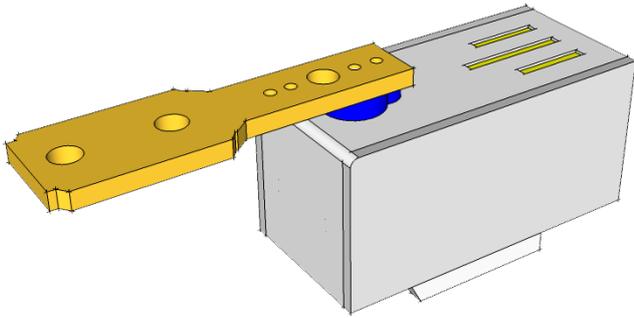
2) Você acha que com a chegada dos carros que dirigem sozinhos, ainda vão ser necessários os semáforos?

3) Você acha que as cidades deveriam dar preferência para os pedestres ou para os carros?

Controlando servo motor

Os servo motores são dispositivos que podemos **gitar** em um ângulo que varia entre **0 e 180 graus**. O professor irá lhe demonstrar como ele funciona. Na sequência faça **a montagem abaixo**.

Fazendo um servo girar



Programação:

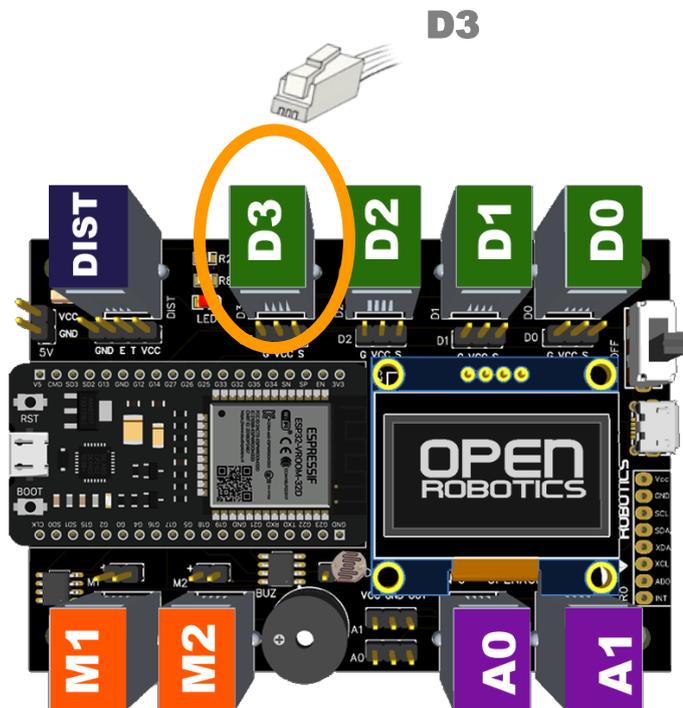
```
#include <darwin6.h>

void setup() {
  ServoNaPorta(D3);
}

void loop() {
  MoverServo(D3, 0);
  EsperarSegundos(2);
  MoverServo(D3, 180);
  EsperarSegundos(2);
}
```

Desafios desta aula:

1. Ligue 2 leds de cores diferentes nas portas D1 e D2. Agora faça com que o LED em D1 fique aceso quando o servo estiver na posição 0 graus, e o LED em D2 fique aceso quando o servo estiver na posição 180 graus.



Olimpíadas de robótica

As Olimpíadas de Robótica são eventos educacionais e competitivos que reúnem equipes de estudantes para competirem em desafios envolvendo a construção e programação de robôs. Essas competições têm como objetivo estimular o interesse em ciência, tecnologia, engenharia e matemática (STEM, na sigla em inglês) entre os participantes, além de promover o trabalho em equipe e habilidades práticas.

Existem várias competições de robótica em todo o mundo, cada uma com suas regras e desafios específicos. Alguns exemplos incluem:

FIRST Robotics Competition (FRC): Uma competição internacional onde equipes de estudantes constroem robôs de grande escala para realizar tarefas complexas em um campo de jogo.

VEX Robotics Competition: Uma competição que envolve a construção de robôs mais compactos, usando peças VEX, para competir em desafios específicos.

RoboCup: Já mencionado anteriormente, a RoboCup é uma competição focada em futebol robótico, onde equipes de robôs autônomos competem em partidas simulando um jogo de futebol.

World Robot Olympiad (WRO): Uma competição internacional que abrange diversas categorias e desafios de robótica para estudantes de diferentes faixas etárias.

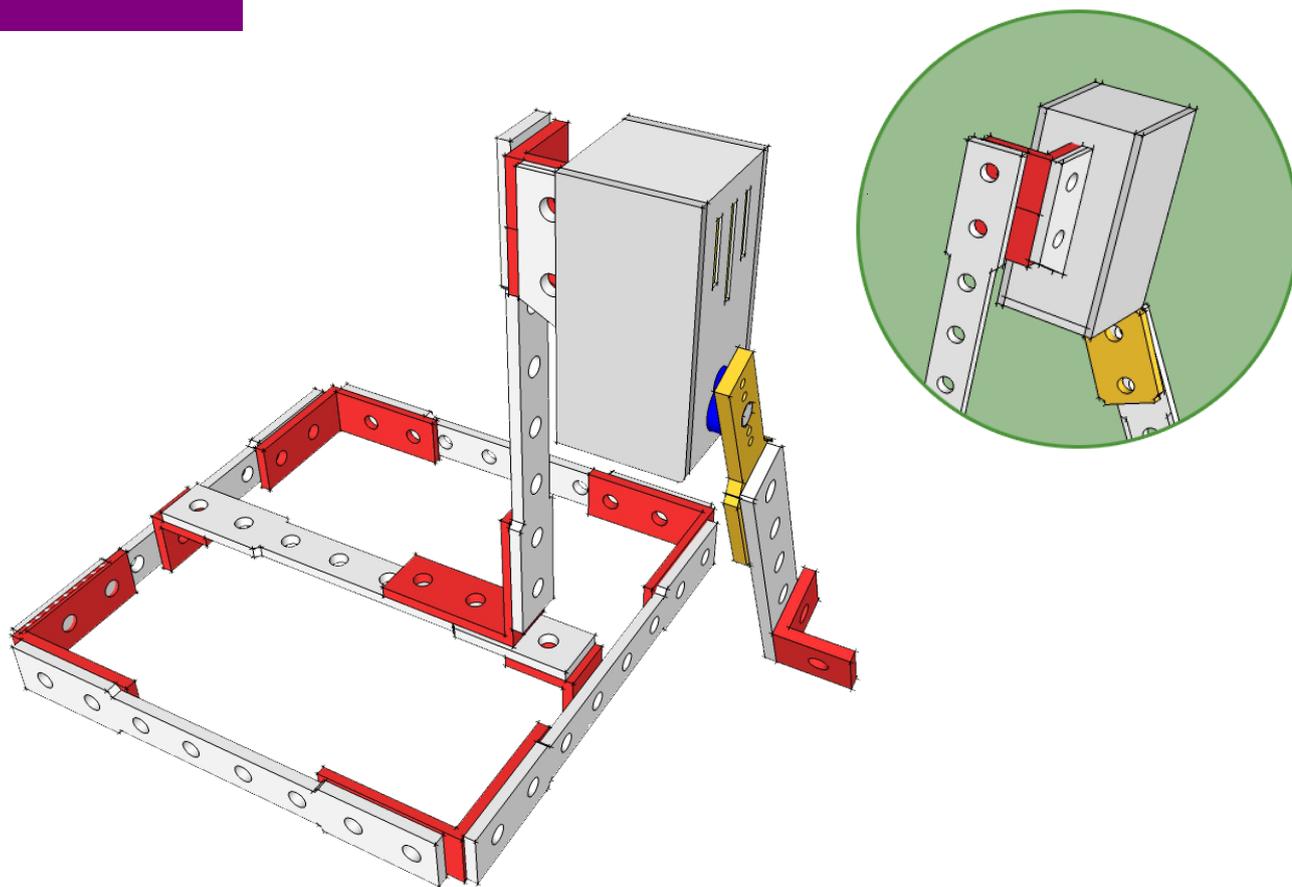
Botball: Uma competição de robótica educacional onde equipes constroem e programam robôs autônomos para realizar tarefas em uma arena.

Nessas competições, os estudantes são desafiados a projetar, construir e programar robôs que possam completar tarefas específicas dentro de um ambiente competitivo. Além do aspecto competitivo, as Olimpíadas de Robótica enfatizam valores como cooperação, aprendizado e resolução de problemas, contribuindo para o desenvolvimento de habilidades importantes para os participantes no campo da tecnologia e engenharia.



Montando um robô chutador

Vamos agora montar um robô que chuta a bola ao gol. Vamos utilizar uma **base** que dará **sustentação** ao nosso robô. Para completar essa montagem, você pode também fazer um gol, que deverá ser acertado pelo jogador.



Você vai precisar dos seguintes materiais: 1 servo motor, 1 botão de toque, 1 placa Darwin, 6 barras 8 furos, 1 barra de 5 furos, 5 conectores L com 2 furos, 1 haste para o servo motor, 5 conectores L 4 furos, 1 bolinha de ping-pong, parafusos e porcas.

Para montar o robô chutador faça o seguinte:

1. Comece montando a **base com 4 peças de 8 furos** e também **4 peças L de 4 furos** nos cantos. Em duas laterais coloque **1 peça L de 2 furos**, sobre as quais você vai fixar **1 barra de 8 furos**. Na peça central, **coloque um L de 4 furos** para conectar uma outra **peça com 8 furos**.
2. No topo desta peça faça a instalação do **servo motor** utilizando uma **peça L com 2 furos**.
3. Conecte **uma haste** no servo motor utilizando o parafuso para servo (**micro parafuso**).
4. Conecte a **barra de 5 furos** na haste do servo e adicione uma **peça L de 2 furos** para ser o pé.
5. Ligue o **botão de push na porta D2** e o **servo motor na porta D3**. O servo fará a movimentação da perna do robô para chutar a bolinha de ping-pong.

Programando o robô chutador

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduíno**.

Programação:

```
#include <darwin6.h>
int ValorDoBotao;
void setup() {
  ServoNaPorta(D3);
  BotaoNaPorta(D2);
}
void loop() {
  MoverServo(D3, 90);
  ValorDoBotao = LerBotao(D2);
  if(ValorDoBotao == Sim)
  {
    MoverServo(D3, 150);
    EsperarSegundos(1);

    MoverServo(D3, 90);
    EsperarSegundos(1);
  }
}
```

Entendendo o que o programa faz:

Na primeira parte criamos a **variável ValorDoBotao** para armazenar a situação em que se encontra o **botão** (apertado ou não), além de iniciar o **servo motor na posição 90 graus**.

Na sequência existe um **teste para verificar se o botão** está pressionado, **em caso afirmativo (Sim) então o servo que está na porta D3 da placa irá se movimentar**. Veja que talvez você tenha que alterar os ângulos, dependendo de como ficou a posição do servo que você instalou na placa. Não se esqueça que o professor está aí para te ajudar.

Ampliando seus conhecimentos

Após a realização da **montagem e da programação** da estrutura do robô chutador, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Porque você acha que o Rugby não é muito difundido no Brasil?

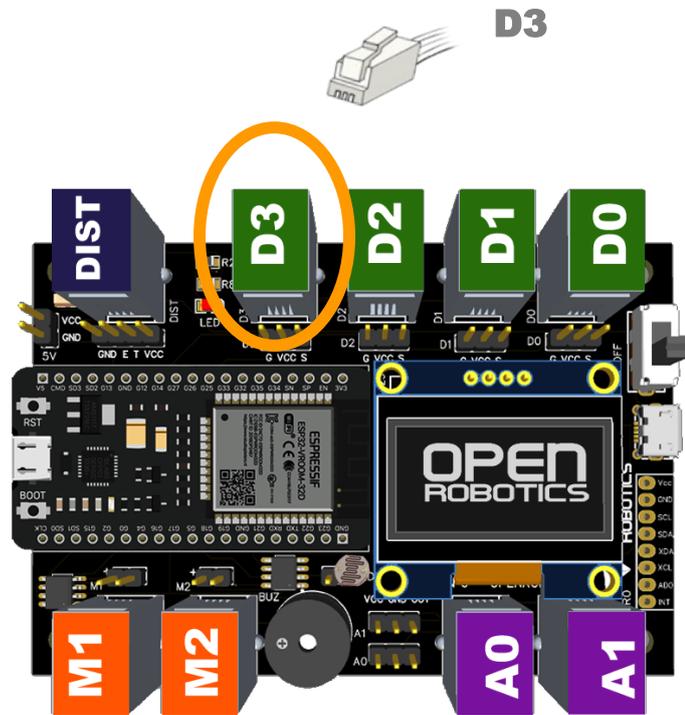
2) Quais as principais regras do futebol?

3) Qual foi o maior jogador de basquete brasileiro de todos os tempos? Cite algumas façanhas dele.

Emitindo sons

O uso de **sirenes** pode se tornar útil quando precisamos emitir algum **sinal sonoro**. Utilizando uma sirene (também conhecida como **Buzzer**) podemos emitir um alerta ou reproduzir diversos tipos de **sons**.

Fazendo a sirene funcionar



Programação:

```
#include <darwin6.h>

void setup() {
    //Informando que a sirene está em D3
    SireneNaPorta(D3);
}

void loop()
{
    LigarSirene(D3);
    EsperarSegundos(2);
    DesligarSirene(D3);
    EsperarSegundos(2); }
```

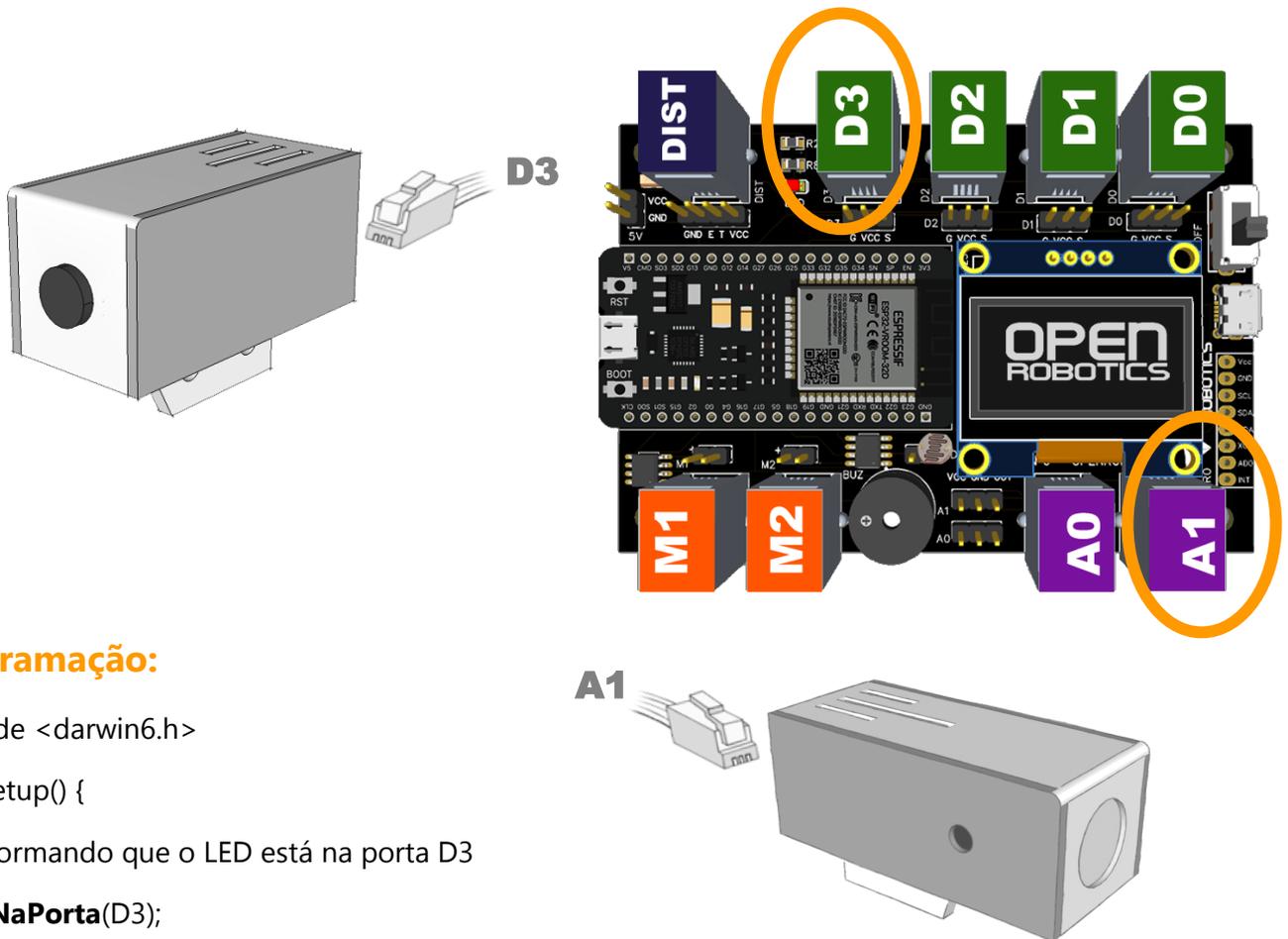
Desafios desta aula:

1. Instale um LED na porta D2 e faça ele ligar quando a sirene estiver desligada.
2. Faça o LED e a sirene tocarem juntos por um intervalo de 1 segundo, sem parar.

Sensores de LUZ ambiente

Agora vamos ver como podemos medir a **intensidade da luz** no ambiente onde estamos. Para isso vamos utilizar um **sensor de luz**, também chamado de **LDR (Light Dependent Resistor)**. Com ele vamos ativar um **LED** quando estiver escuro e apagaremos o LED quando estiver claro, **simulando** o que acontece com o poste de luz nas ruas.

Acionando o LED pelo LDR



Programação:

```
#include <darwin6.h>

void setup() {
    //Informando que o LED está na porta D3
    LedNaPorta(D3);
}

void loop() {
    if ( LerLuz(A1) == Escuro )
        LigarLed(D3);
    else
        DesligarLed(D3);
}
```

Desafios desta aula:

1. Instale mais um LED na porta D1. Faça com que ele acenda quando o ambiente estiver iluminado.
2. Utilizando um servo na porta D2, faça com que ele gire para a posição 0 graus quando estiver escuro e 180 graus quando estiver claro.

Robótica salvando vidas

Sim, os robôs desempenham um papel significativo em salvar vidas em diversas áreas. Algumas das aplicações em que robôs são utilizados para salvamento e assistência incluem:

Robôs de Busca e Resgate: Em situações de desastres naturais, como terremotos, incêndios ou inundações, robôs podem ser enviados para áreas de difícil acesso para procurar sobreviventes e fornecer informações cruciais às equipes de resgate.

Robôs Médicos: Na área da saúde, robôs são usados em cirurgias assistidas, permitindo procedimentos mais precisos e menos invasivos. Alguns robôs também são utilizados em ambientes hospitalares para a entrega segura de medicamentos e outros materiais.

Veículos Autônomos para Transporte de Emergência: Robôs autônomos, como drones e veículos terrestres, podem ser usados para transportar suprimentos médicos, amostras de sangue, ou até mesmo órgãos para transplante, acelerando o processo e garantindo entregas rápidas em situações críticas.

Robôs de Desativação de Bombas: Em situações de ameaças explosivas, robôs especializados são frequentemente usados para desativar explosivos de forma segura, protegendo as equipes de intervenção humana de riscos potenciais.

Robôs para Detecção de Substâncias Perigosas: Robôs equipados com sensores especializados podem ser usados para detectar substâncias perigosas ou radioativas em áreas de risco, permitindo uma resposta mais segura e eficiente.

Robôs de Monitoramento Ambiental: Em situações de desastres ambientais, como vazamentos de produtos químicos, robôs podem ser usados para monitorar a qualidade do ar, da água e do solo, ajudando na avaliação dos danos e na tomada de decisões informadas. Esses são apenas alguns exemplos, e a tecnologia continua avançando, expandindo as possibilidades de como os robôs podem ser utilizados para salvar vidas e melhorar a eficiência em situações de emergência e resgate.



Simulando a iluminação pública

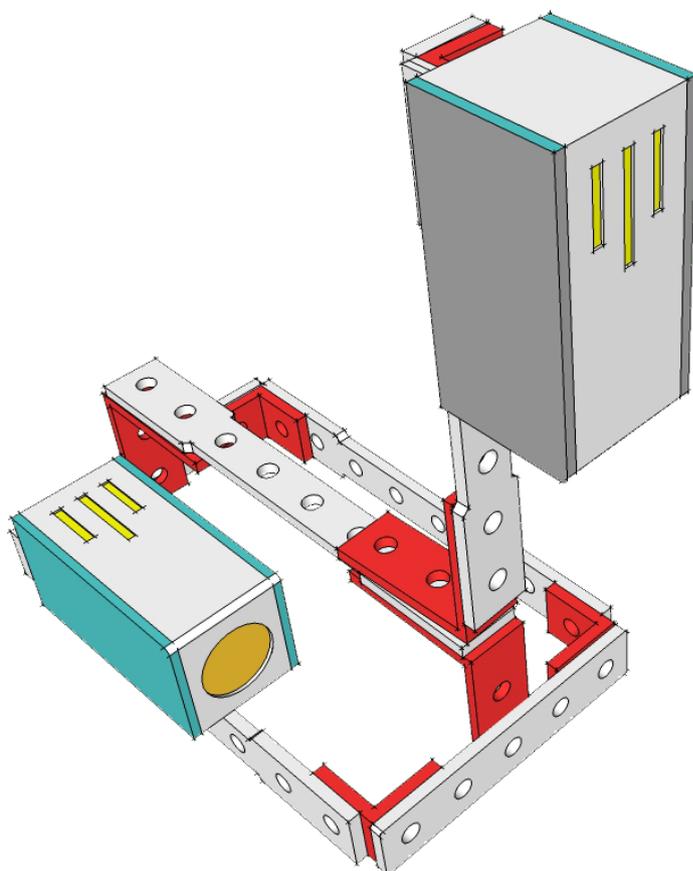
Uma das formas que os **habitantes do mundo** podem contribuir é justamente economizando energia, uma vez que para produzir energia elétrica, de uma maneira geral, **poluímos muito o meio em que vivemos**.

Podemos, por exemplo, **utilizar sensores** de luz em vias públicas, que farão com que as luzes dos postes se acendam apenas quando a noite chegar, ou quando o tempo estiver muito fechado durante o dia.

Vamos agora ver uma montagem simulando o sistema de iluminação pública, em que as luzes se acendem ao escurecer.

Você vai precisar dos seguintes materiais:

1 sensor de Luz, 1 lâmpada LED, 4 barras 8 furos, 3 conectores L 4 furos, 2 barras de 5 furos, 5 conectores L 2 furos, placa Darwin, parafusos e porca, papelão ou papel A4.



Para montar o sistema de iluminação faça o seguinte:

1. Comece montando a **base com 3 peças de 8 furos, 2 barras de 5 furos e 4 peças L de 2 furos** nos cantos. Na peça central, coloque **3 peças L 4 furos e 1 barra de 8 furos**.
2. Para o **poste**, utilize **1 barra L 8 furos**, e no alto dela instale o **LED utilizando 1 peça L 2**. Ele vai funcionar como as luminárias que estão no alto dos postes.
3. Instale o **LED na porta D2** da placa Darwin. Instale o **sensor de luz na porta A1**.
4. Quando a incidência de luz no sensor estiver baixa (escuro) o LED deve acender. Você pode desenhar e forrar a base com o **papel A4 simulando uma avenida**.

Programando o sistema de iluminação

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**.

1. Na área de trabalho do seu computador dê dois cliques no ícone do **Arduino**.

2. Então crie o programa mostrado abaixo:

Programação:

```
#include <darwin6.h>

int x;

void setup() {
  LedNaPorta(D2);
}

void loop()
{
  x = LerLuz(A1);
  if ( x == Escuro )
    LigarLed(D2);
  else
    DesligarLed(D2);
}
```

Entendendo o que o programa faz:

Após instalado o **LED na porta D2**, fazemos a leitura da porta onde está o **sensor de Luz (A1)**, e colocamos esse valor na **variável x** que foi definida para isso.

SE a luz que entra pelo orifício do sensor estiver **fraca (Escuro)**, então será dado o comando para **acender o LED**, caso contrário, ou seja, se a iluminação estiver **média ou alta (Claro)**, o **LED permanecerá apagado**.

Desafio:

Altere a montagem e também a programação para que haja 2 postes de luz na avenida.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** da estrutura do sistema de iluminação pública, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Como é a iluminação pública em sua cidade?

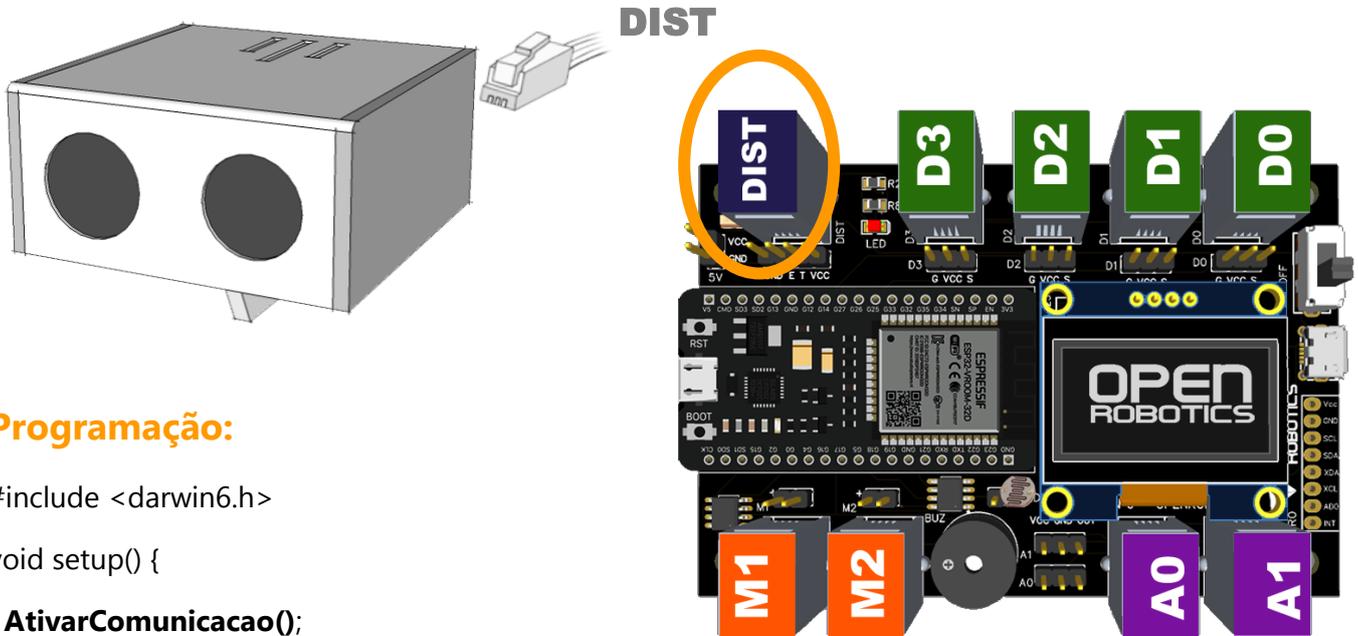
2) Você acha que um sensor como este poderia ser utilizado em sua residência?

3) Qual a importância de se economizar energia?

Sensores de distância

Alguns robôs possuem **sensores ultrassônicos** que são utilizados para medir a distância dos objetos. Este recurso torna-se muito útil quando ele precisa **desviar de obstáculos** ou seguir algum objeto até que o encontre em um **determinado ambiente**. Veremos agora como isso pode ser feito com o sensor de distância por ultrassom.

Medindo a distância em centímetros



Programação:

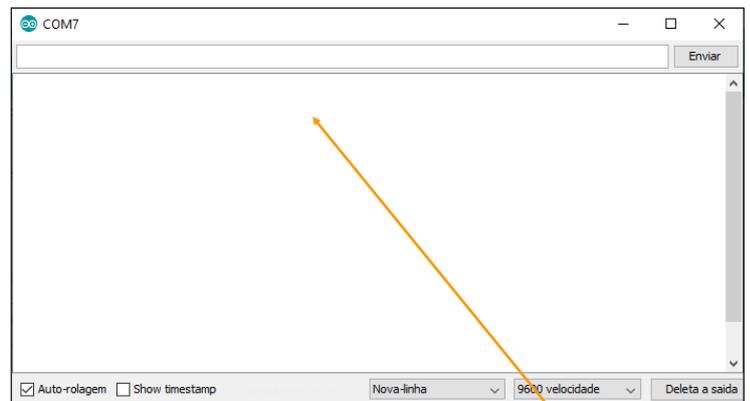
```
#include <darwin6.h>

void setup() {
  AtivarComunicacao();
}

void loop() {
  EnviarParaComputador ( LerDistancia() );
  EsperarSegundos(1);
}
```

Novidade na área:

Perceba que no programa acima utilizamos duas novas funções: **AtivarComunicacao()** e **EnviarParaComputador()**. Com elas é possível receber e **monitorar o valor sendo lido por um sensor em tempo real**. Para isso, enquanto o programa estiver em execução, clique no **Ferramentas** do programa do Arduino e acione a opção **Monitor Serial**. Ao aproximar ou distanciar um objeto plano da frente do sensor, a distância será exibida **nesta área**. Experimente!



Desafios desta aula:

1. Instale um LED na porta D2. Faça com que ele acenda SE a distância for menor que 10 cm.
2. Instale a sirene na porta D1. Faça com que ela dê 1 apito quando a distância for menor que 10 cm; 2 apitos quando a distância estiver entre 11 e 20 cm e 3 apitos quando a distância estiver entre 21 e 30 cm.

Insetos robôs

Os insetos robôs, ou robôs inspirados em insetos, são uma área de pesquisa em robótica biomimética, onde os projetos de robôs são modelados com base nas características e comportamentos de insetos reais. Esses robôs são frequentemente projetados para imitar a locomoção, a capacidade de voo e outras características específicas dos insetos. Aqui estão alguns pontos de destaque sobre insetos robôs:

Locomoção: Muitos insetos robôs são projetados para replicar os modos de locomoção de insetos reais, como andar, correr ou escalar superfícies. Isso é útil em aplicações onde a mobilidade em ambientes complexos é necessária.

Voo: Alguns robôs são desenvolvidos para imitar a capacidade de voo de insetos. Isso pode incluir o uso de asas mecânicas ou hélices para propulsão, o que é especialmente útil em áreas como exploração aérea ou monitoramento ambiental.

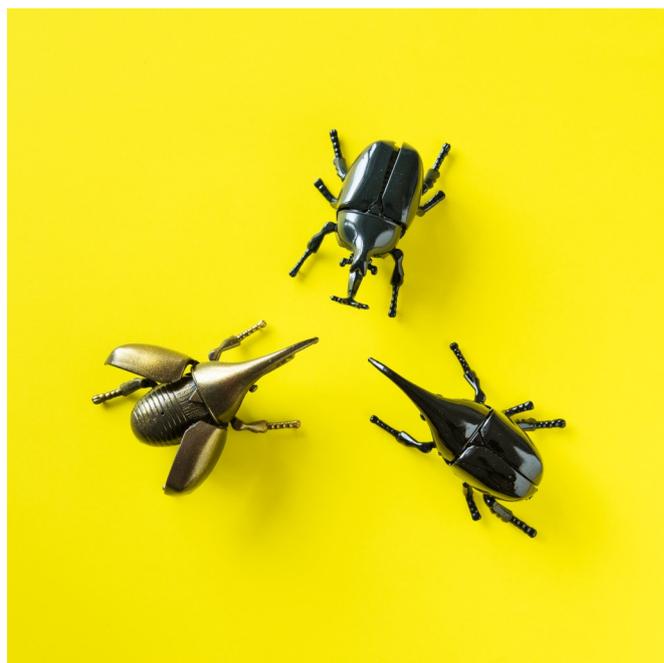
Aplicações de Vigilância e Exploração: Insetos robôs podem ser empregados em aplicações de vigilância e exploração, onde sua capacidade de locomoção ágil e tamanho compacto podem ser vantajosos. Eles podem ser usados em ambientes de difícil acesso, como espaços urbanos confinados ou áreas de desastres.

Agricultura: Algumas pesquisas exploram o uso de insetos robôs na agricultura, onde podem ser usados para polinização de plantas ou monitoramento de culturas.

Medicina: Insetos robôs também têm potencial aplicação na medicina, como em procedimentos minimamente invasivos, como a entrega de medicamentos em áreas específicas do corpo.

Robôs Colaborativos em Enxames: Assim como os insetos trabalham em enxames, os robôs podem ser projetados para colaborar em grandes grupos para realizar tarefas complexas. Cada robô individual atua de forma autônoma, respondendo ao ambiente e às ações dos outros.

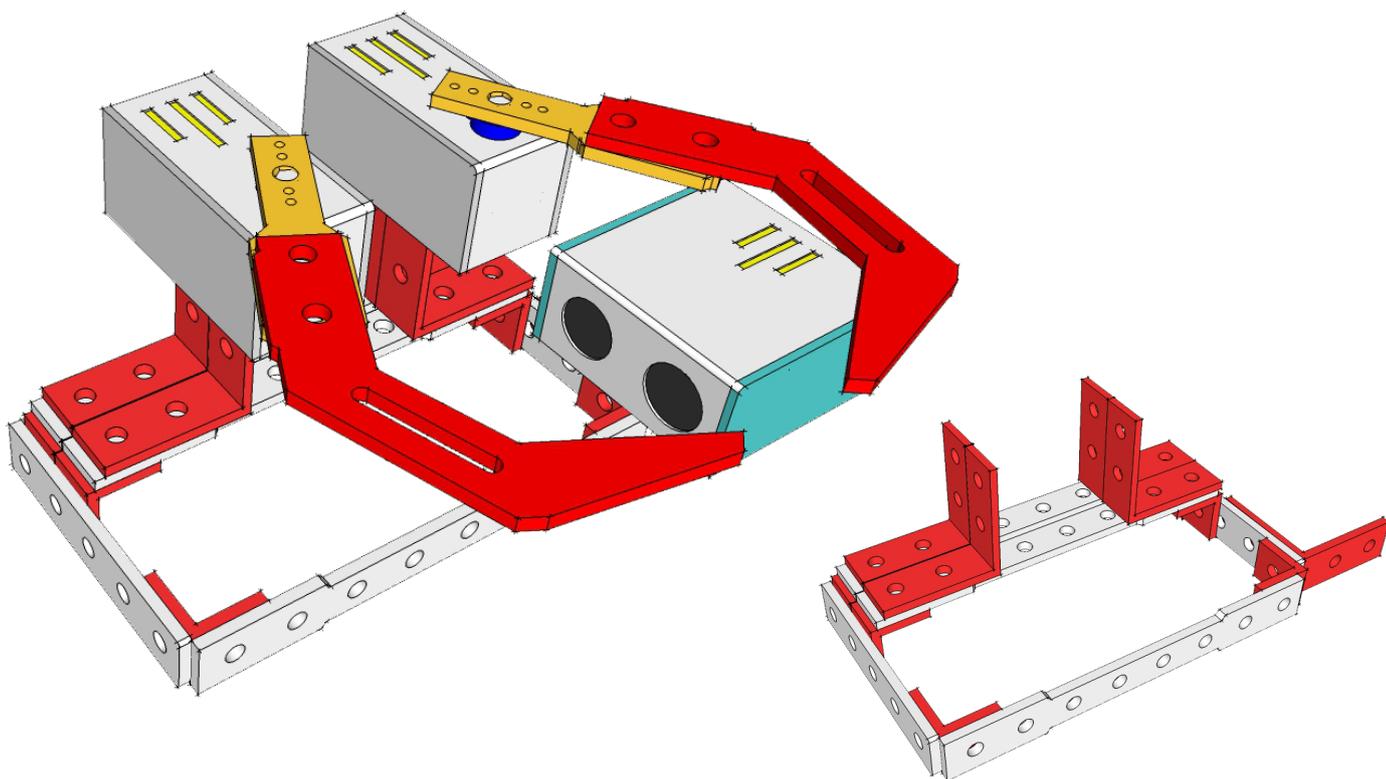
A biomimética, em geral, busca inspiração na natureza para desenvolver soluções inovadoras e eficientes em diversos campos, e os insetos robôs representam uma abordagem fascinante nesse sentido. É importante notar que, embora existam avanços significativos, a implementação prática desses conceitos ainda está em estágios de pesquisa e desenvolvimento.



Montando uma garra robótica

Você já deve ter notado que formiga, escorpião, gafanhoto, alguns besouros e vários outros insetos possuem uma ou mais pinças em alguma parte do corpo. A utilidade dessa garrinha para esses bichinhos deve ser enorme! Pense nas formigas cortando as folhas por exemplo...

Talvez por inspiração, alguns robôs são montados tendo como mãos as pinças, que alguns também chamam de garras. Vamos agora montar a garra de um robô.



Você vai precisar dos seguintes materiais:

1 sensor de ultrassom (distância), 2 servos motores, 2 garras, 5 conectores L 4 furos, 6 conectores L 2 furos, 3 barras longas B 8 furos, 2 barras pequenas B 5 furos, 2 hastes para os servos, a placa Darwin, parafusos e porcas.

Para montar a garra robótica faça o seguinte:

1. Comece montando a **base com 2 barras pequenas de 5 furos e 1 barra longa de 8 furos conectadas por 2 peças L de 2 furos**. Acrescente **1 peça L de 4 furos** para ficar fora da base. Na outra lateral longa da base acrescente **2 peças L de 2 furos juntas** em cada extremidade. Sobre essas 4 peças L de 2 furos, você fixa as **2 barras de 8 furos** restantes.
2. Note no desenho que os **4 L de 4 furos** estão de forma que sejam **o apoio dos servo motores**.
3. Nos eixos dos servos parafuse as **hastes** usando os **micro parafusos**. Então adiciona **uma garra** em cada. Depois coloque o **sensor de distância** em outra lateral da base.
4. Os 2 servos motores deverão ser ligados nas **portas D3 e D2**. O sensor de distância deve ser ligado na **porta DIST**.

Programando a garra

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduíno**. Na área de trabalho do seu computador dê dois cliques no ícone do **Arduíno**. Então crie o programa mostrado abaixo:

Programação:

```
#include <darwin6.h>

int Distancia;

void setup()
{
  ServoNaPorta(D3);
  ServoNaPorta(D2);
}

void loop()
{
  Distancia = LerDistancia();
  if(Distancia < 20) {
    MoverServo(D3, 60);
    MoverServo(D2, 110);
    EsperarSegundos(5);
  }
  else {
    MoverServo(D3, 180);
    MoverServo(D2, 0);
  }
}
```

Entendendo o que o programa faz:

Esse programa faz com que as garras se fechem automaticamente toda vez que algo se aproximar do sensor de distância. No início pegamos as informações do sensor de distância conectado na **porta DIST** e passamos para a **variável Distancia**. Caso algo esteja próximo ao sensor de distância, os **servos motores (garras)** são posicionados em determinados ângulos para se fecharem. Então **aguardamos 5 segundos**, como se a garra estivesse segurando algo. Caso não tenha nada próximo, os servos motores são posicionados de maneira que as garras fiquem abertas.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** da estrutura da garra robótica, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Quais insetos mais te deixam com medo? Por quê?

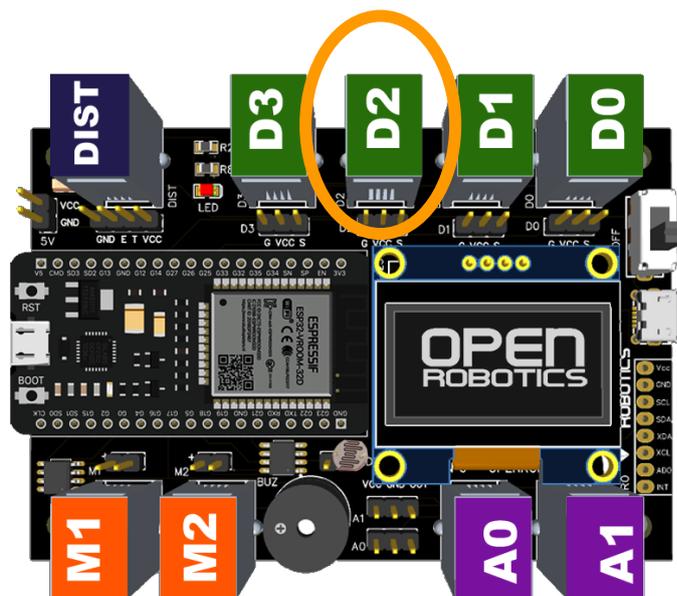
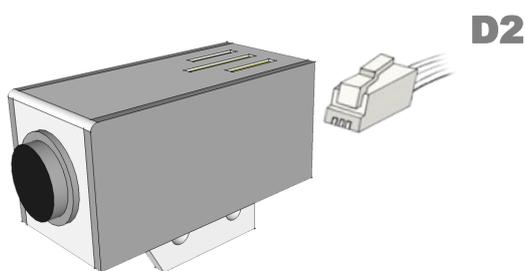
2) Cite alguns usos que os insetos podem dar para suas garras em forma de pinças, parecidas com a que montamos.

3) Pesquise na Internet sobre o mosquito *Aedes Aegypti* e escreva alguns cuidados que podemos tomar para evitar sua proliferação nos meios urbanos.

Acionando botão push button

Em diversas situações é importante contar com um **botão** para que uma pessoa possa apertar e disparar **alguma ação no robô**. Por exemplo, podemos utilizar um botão para enviar um sinal para que o robô possa rodar ou parar um motor; **ligar ou desligar um LED** e em dezenas de outras situações. Vejamos como isso pode ser feito.

Acionando a sirene da placa Darwin com o botão



Programação:

```
#include <darwin6.h>

int x; //Criamos uma variável para
//ver a situação do botão

void setup() {
  BotaoNaPorta(D2);
  SireneNaPorta(D13);
}

void loop() {
  x = LerBotao(D2);
  if (x == Sim)
    LigarSirene(D13);
  else
    DesligarSirene(D13);
}
```

Desafios desta aula:

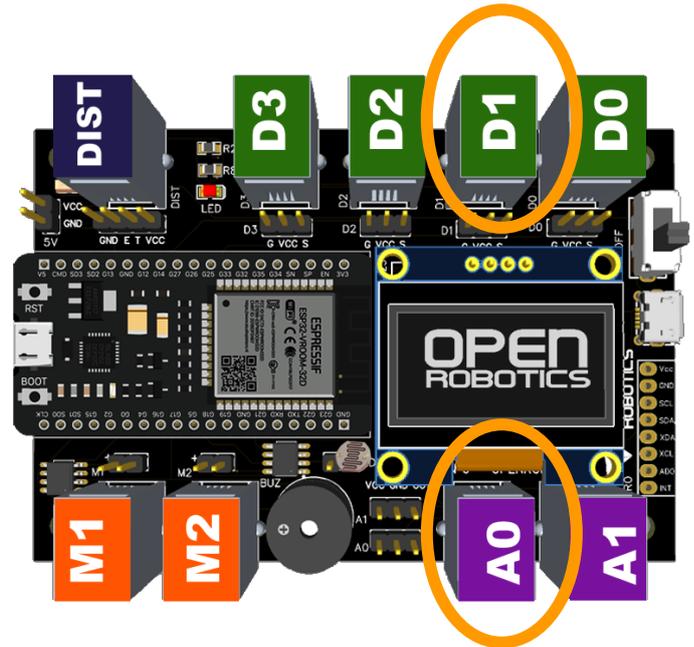
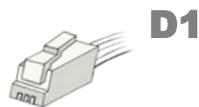
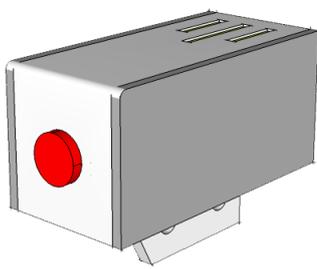
1. Agora use o LED da placa Darwin que está na **porta D12**. Caso o botão esteja pressionado esse LED deverá ser ligado.
2. Instale um servo motor na **porta D3**. Caso o botão seja pressionado, então o servo deverá girar para o ângulo 0 graus. Caso ele esteja solto, então o servo deverá girar até 180 graus.

Sensor de Infravermelho

Um sensor de **infravermelho** pode ser utilizado quando precisamos medir uma luz que não pode ser vista **pelo ser humano**. Os raios infravermelhos possuem diversos usos no nosso dia a dia, por exemplo, são utilizados em controle remotos de **aparelhos eletrônicos**.

Para esta aula você deverá possuir uma superfície preta e outra branca para testar o sensor de infravermelho, o qual deverá estar a uma distância de 1 cm destas superfícies. Vejamos como funciona.

Acionando um LED pela leitura do infravermelho



Programação:

```
#include <darwin6.h>

int x; //Criamos uma variável para ver a
situação do infravermelho

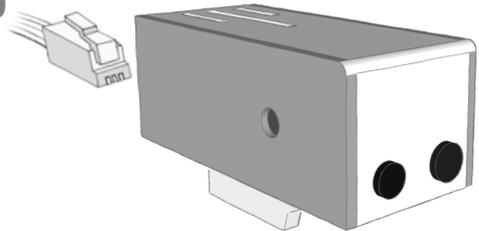
void setup() {
  LedNaPorta(D1);
}

void loop() {
  x = LerInfra(A0);
  if (x == Preto)
    LigarLed(D1);
  else
    DesligarLed(D1);
}
```

Desafios desta aula:

1. Ligue um outro LED na porta D2 que deverá ficar aceso quando o Infravermelho estiver sobre a superfície branca.
2. Acione a sirene da placa Darwin quando o LED em D2 estiver ligado.

A0



Engrenagens das máquinas

As engrenagens são dispositivos mecânicos compostos por rodas dentadas que se encaixam entre si. Elas são utilizadas para transmitir movimento e potência entre eixos, permitindo a variação de velocidade e torque em sistemas mecânicos. Aqui estão alguns conceitos importantes relacionados a engrenagens:

Dentes: As rodas dentadas têm dentes que se encaixam, permitindo a transferência eficiente de movimento e força. A forma dos dentes, bem como o número de dentes, é projetada para otimizar a transmissão.

Módulo e Passo: O módulo e o passo são medidas que descrevem as características dimensionais de uma engrenagem. O módulo é usado em sistemas métricos, enquanto o passo é usado em sistemas imperiais. Eles indicam a distância entre os dentes.

Relação de Engrenagem: A relação de engrenagem é a relação entre o número de dentes em duas engrenagens interconectadas. Ela determina a relação de velocidade e torque entre as engrenagens.

Engrenagem Reta e Engrenagem Helicoidal: Engrenagens retas têm dentes que são paralelos ao eixo da engrenagem, enquanto engrenagens helicoidais têm dentes em um ângulo em espiral ao redor do eixo. Engrenagens helicoidais oferecem uma transmissão mais suave e silenciosa.

Engrenagem Cônica: Engrenagens cônicas têm dentes que estão dispostos ao longo de um cone. Elas são frequentemente usadas para transmitir movimento entre eixos que se cruzam.

Engrenagem de Sem-fim: Engrenagens de sem-fim consistem em um par de rodas dentadas onde uma tem dentes helicoidais e a outra tem uma superfície roscada (sem-fim). Elas proporcionam uma grande redução de velocidade.

Engrenagem Planetária: Nesse tipo de engrenagem, um conjunto central, chamado de "sol", é cercado por engrenagens menores (planetas) que giram ao seu redor. Engrenagens planetárias são comumente usadas em sistemas de transmissão automotiva e em caixas de marcha.

Sentido de Rotação: A direção do movimento transmitido pelas engrenagens depende da orientação relativa dos dentes. Engrenagens podem ser usadas para inverter ou manter a direção do movimento.

Engrenagens são fundamentais em uma variedade de máquinas e dispositivos, desde motores e transmissões de veículos até relógios e equipamentos industriais. A escolha do tipo correto de engrenagem e sua configuração é crucial para otimizar o desempenho e a eficiência de um sistema mecânico.



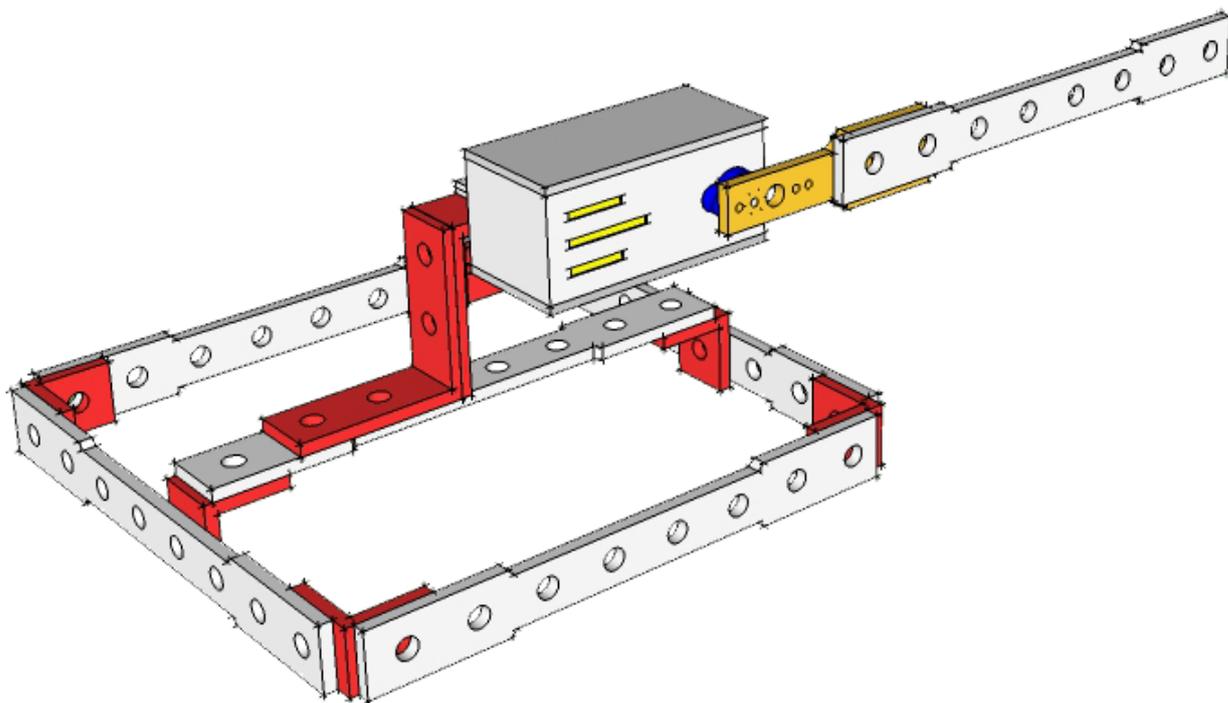
Montando uma cancela automática

Há poucas décadas atrás não era assim, mas hoje **tem computador em tudo quanto é lugar que a gente vai**.

No escritório, na padaria, na lanchonete, no restaurante, na fábrica, na farmácia, no açougue, na igreja, no supermercado, nas lojas, nos bancos, na secretaria da escola e, algumas vezes, **até na sala de aula**.

Na cabine dos pedágios em rodovias também tem um computador para que o atendente possa registrar o pagamento e dar o **comando para levantar a cancela**. Será que o **computador levanta automaticamente a cancela** após a confirmação do pagamento?!

A seguir vamos montar e programar uma cancela automática para que você tenha a experiência com esse mecanismo. Acompanhe as orientações abaixo:



Você vai precisar dos seguintes materiais: 6 barras 8 furos, 1 servo motor, 1 haste do servo motor, 6 conectores L 2 furos, 2 conectores L 4 furos, placa Darwin, parafusos e porcas.

Para montar a cancela faça o seguinte:

1. Comece montando a **base com 5 peças de 8 furos e 6 peças L de 2 furos**. Na peça central, coloque os **2 conectores L 4 furos** conforme o desenho acima.
2. Então acople o **servo motor ao conector L 4 furos** e fixe também sua **haste** com o parafuso apropriado. Acople **a barra de 8 furos à haste** do servo motor. Ela servirá como **braço da cancela**. Conecte o servo motor na **porta D2** da placa Darwin.

Programando a cancela

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduíno**.

1. Na área de trabalho do seu computador dê dois cliques no ícone do **Arduíno**.
2. Então crie o programa mostrado abaixo:

Programação:

```
#include <darwin6.h>

void setup() {
  ServoNaPorta(D2);
}

void loop() {

  MoverServo(D2, 0);
  EsperarSegundos(2);
  MoverServo(D2, 90);
  EsperarSegundos(2);
}
```

Entendendo o que o programa faz:

O segredo nesse programa é fazer com que o **servo motor** se posicione em lugares diferentes para ser feito o **movimento de uma cancela automática**.

Primeiramente posicionamos o **servo na posição 0**, depois **esperamos 2 segundos** para o próximo movimento.

Neste ponto talvez seja necessário você **desparafusar a haste de acrílico do servo motor** e fazer algumas experiências para **descobrir onde fica a posição zero** em relação à sua montagem.

No próximo movimento iremos **abrir a cancela e esperar mais 2 segundos**. Por isso posicionamos o eixo do motor no **ângulo 90**.

Caso a cancela não esteja abrindo adequadamente, **experimente mudar o ângulo de 90** para outros ângulos mais próximos até achar um que mais lhe agrade.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** da cancela, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

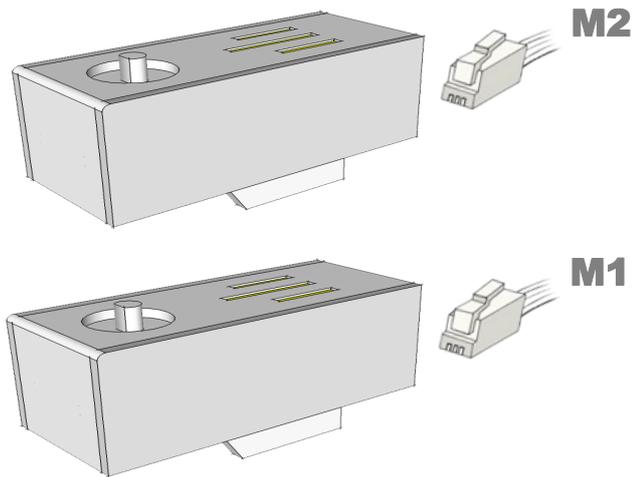
1) Você acha interessante levar o uso dos computadores às pessoas que não podem comprar um? Que ideia você sugere para viabilizar isso?

2) Há alguém na sua família que tem dificuldades ao utilizar tecnologias como o computador, smartphone e a Internet? O que você faz para ajudar?

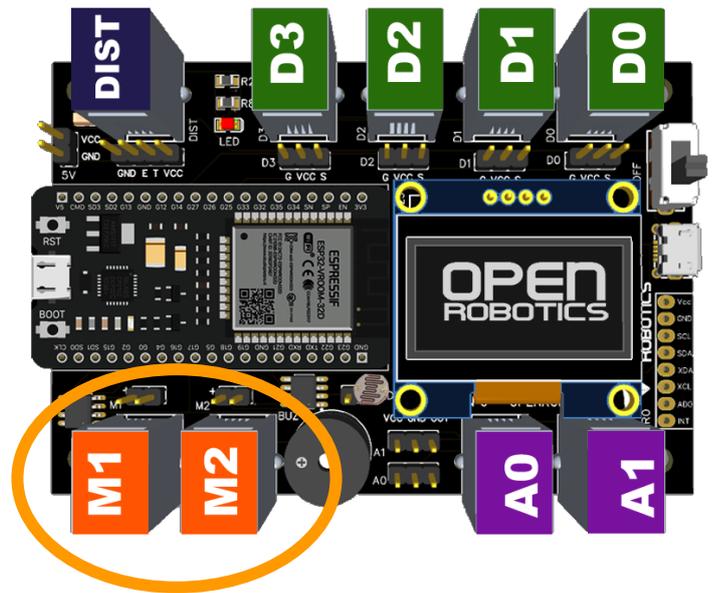
3) Quais as baterias de computador mais econômicas e sustentáveis? (Pesquise na Internet)

Acionando motores de Corrente Contínua (CC)

Os **motores** são dispositivos muito utilizados na robótica. Eles podem ser de vários tipos e formatos. Para que eles funcionem é necessário ligar um dispositivo **conhecido como ponte H**, o qual já vem embutido na placa **Darwin da DUOBIT**. Ela possui duas **entradas específicas para os motores CC** que estão com os nomes de **M1** e **M2**. Vejamos como funciona.



Acionando motores CC



Programação:

```
#include <darwin6.h>
void setup() {
  AtivarMotores();
}
void loop() {
  MotorM1(Frente);
  MotorM2(Frente);
  EsperarSegundos(1);
  MotorM1(Parar);
  MotorM2(Parar);
  EsperarSegundos(1);
  MotorM1(Re);
  MotorM2(Re);
  EsperarSegundos(1);
}
```

O que é a Inteligência Artificial

A Inteligência Artificial (IA) refere-se à capacidade de um sistema ou máquina de realizar tarefas que normalmente exigiriam inteligência humana. Essas tarefas incluem aprendizado, raciocínio, reconhecimento de padrões, processamento de linguagem natural, resolução de problemas e tomada de decisões. Aqui estão alguns conceitos-chave relacionados à inteligência artificial:

Aprendizado de Máquina (ML): É um subcampo da inteligência artificial que envolve o desenvolvimento de algoritmos e modelos que permitem que um sistema aprenda padrões a partir de dados. Existem duas principais categorias de aprendizado de máquina: supervisionado (onde o modelo é treinado com dados rotulados) e não supervisionado (onde o modelo tenta encontrar padrões por conta própria).

Redes Neurais: Inspiradas no funcionamento do cérebro humano, as redes neurais são estruturas que consistem em camadas de neurônios artificiais. Elas são amplamente utilizadas em tarefas de aprendizado profundo (deep learning).

Processamento de Linguagem Natural (PLN): Refere-se à capacidade de uma máquina entender, interpretar e gerar linguagem humana. Isso inclui tarefas como tradução automática, análise de sentimentos, chatbots e reconhecimento de fala.

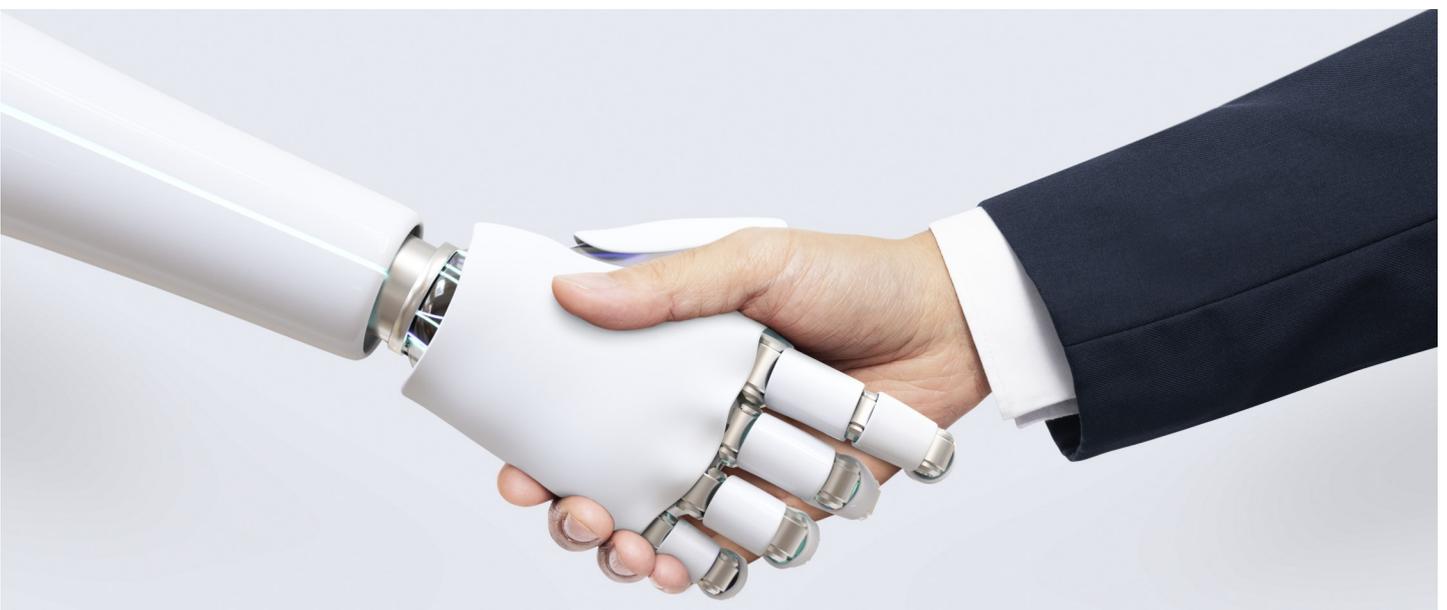
Visão Computacional: É a capacidade de um sistema interpretar e compreender informações visuais. Isso inclui o reconhecimento de objetos, detecção de padrões em imagens e segmentação de objetos.

Sistemas de Recomendação: Utilizados em plataformas online, esses sistemas analisam padrões de comportamento do usuário para sugerir produtos, serviços ou conteúdo relevante.

Agentes Autônomos: São sistemas de IA capazes de tomar decisões e realizar ações de forma autônoma. Exemplos incluem carros autônomos, drones e robôs industriais.

Ética em IA: Com o avanço da inteligência artificial, questões éticas tornaram-se proeminentes, incluindo preocupações com preconceito algorítmico, privacidade, transparência e o impacto da automação no emprego.

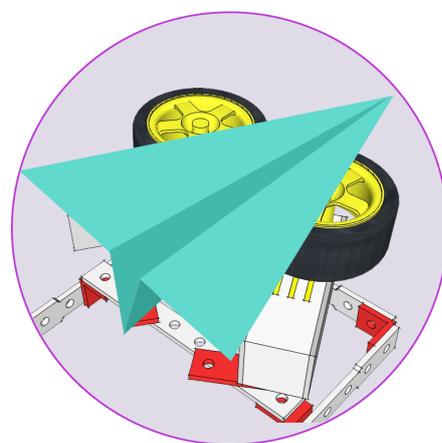
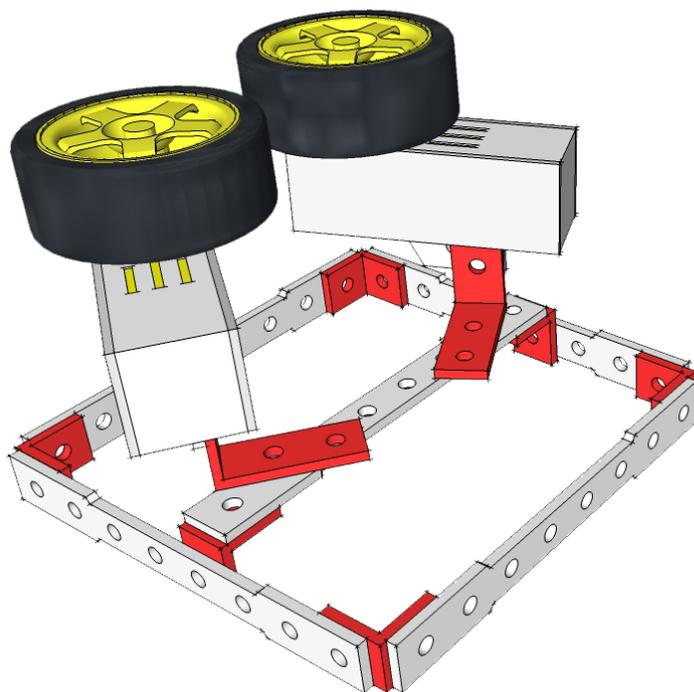
IA Fraca e IA Forte: A IA fraca (ou estreita) refere-se a sistemas especializados em tarefas específicas, enquanto a IA forte (ou geral) seria uma forma de inteligência que pode entender, aprender e aplicar conhecimentos em uma ampla variedade de domínios, comparável à inteligência humana.



Montando um lançador de aviões de papel

Quando **lançamos um corpo verticalmente para cima** notamos que ele sobe até uma certa altura e depois cai, porque é **atraído pela Terra**. O mesmo acontece quando soltamos um corpo de determinada altura. Os corpos são atraídos pela Terra porque em torno dela há uma região chamada **campo gravitacional** exercendo atração sobre eles.

Assim sendo, se criarmos um **lançador robótico de aviões de papel**, em algum momento o aviãozinho vai pousar, ou melhor, cair. Vamos experimentar isso:



Você vai precisar dos seguintes materiais: 2

motores CC, 2 conectores L 4 furos, 6 conectores L 2 furos, 5 barras B 8 furos, 2 rodas de borracha, 1 avião de papel, 1 botão de toque, placa Darwin, parafusos e porcas.

Para montar o lançador de aviões faça o seguinte:

1. Para esta montagem vamos ligar **dois motores em uma base com 5 barras de 8 furos conectadas por 6 peças L de 2 furos**.
2. Perceba pela imagem que as **peças em L 4 furos precisam formar um ângulo** de forma que as duas rodas se encostem, para que crie uma região de atrito onde o avião será posicionado.
3. A ideia é fazer com que, ao ligar os motores, eles possam ejetar o avião com uma força relativamente baixa, mas que possa lança-lo ao ar.
4. Talvez você irá **precisar de uma gominha** para fazer com que os dois motores fiquem pressionados um contra o outro.
5. Conecte um **botão push na porta D2**.
6. Conecte os motores cc **nas portas M1 e M2**. Utilize pilhas para essa montagem.

Programando o lançador de aviões

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**.

Na área de trabalho clique no programa **Arduino**. Então crie o programa mostrado abaixo:

Programação:

```
#include <darwin6.h>

int ValorDoBotao;

void setup() {
  AtivarMotores();
  BotaoNaPorta(D2);
}

void loop() {
  ValorDoBotao = LerBotao(D2);
  if (ValorDoBotao == Sim){
    MotorM1(Frente);
    MotorM2(Re);
  }
  else{
    MotorM1(Parar);
    MotorM2(Parar);
  }
}
```

Entendendo o que o programa faz:

Logo após coletarmos **o estado do botão na variável ValorDoBotao**, perguntamos **SE** ele está **pressionado (Sim)**. Caso esteja, vai começar a **girar os motores CC em sentido contrário** para que o avião de papel, que está pressionado entre as rodas, seja lançado ao ar.

Caso tenha dificuldade em fazer os motores girarem em sentidos opostos, experimente **inverter as opções Frente e Re**.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** do lançador de aviões, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Pesquise na Internet sobre os estudos para uma visita ao planeta Marte. Anote aqui embaixo o que encontrar de mais interessante.

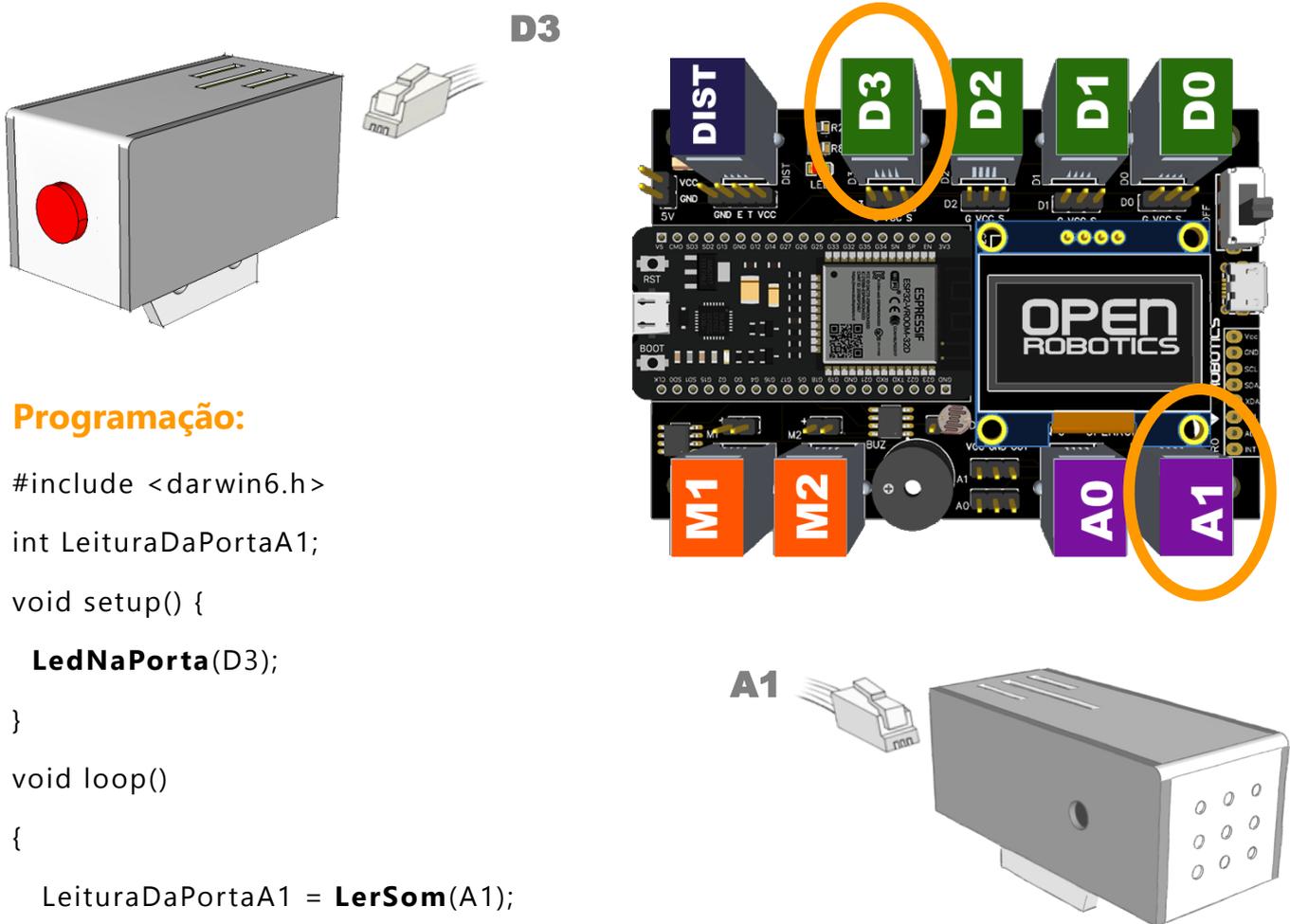
2) Qual é a composição dos anéis que circundam o planeta Saturno? (Pesquise na Internet)

3) Se você pudesse escolher um dos demais planetas do sistema solar para visitar, qual seria? Justifique.

Conhecendo os sensores de SOM

Os **sensores de som** são dispositivos muito importantes na robótica. Eles são capazes de detectar determinados tipos de sons no ambiente. Nesta aula vamos ver como **podemos ligar um LED** quando batemos o **microfone do sensor detectar um som**. Vejamos como ele funciona.

Ligando um LED com o som



Programação:

```
#include <darwin6.h>
int LeituraDaPortaA1;
void setup() {
  LedNaPorta(D3);
}
void loop()
{
  LeituraDaPortaA1 = LerSom(A1);
  if ( LeituraDaPortaA1 == Sim)
    LigarLed(D3);
  else
    DesligarLed(D3);
}
```

Desafios desta aula:

1. Altere o programa anterior para ligar um outro LED na porta D2 quando o ambiente estiver em silêncio.
2. Instale uma sirene na porta D1. Ela deverá ser acionada quando o sensor detectar algum som.

Tecnologias do Futuro

Tecnologia é um termo muito amplo, mas podemos resumir como sendo o **produto final da aplicação de determinados conhecimentos científicos**. Muitas áreas estão envolvidas com o desenvolvimento tecnológico, algumas delas são: medicina, arquitetura, biologia, engenharia, química, indústria e nutrição.

Certas informações podem parecer irrelevantes, mas por mais que sejam óbvias despertam a curiosidade de todos aqueles que gostam de tecnologia. Veja algumas **tecnologias que podem revolucionar as nossas vidas** no futuro:

1) Drones como método de entrega: atualmente o **maior problema** para que os drones finalmente comecem a fazer entregas é a **legislação**, mas os futuristas apontam que em breve eles serão usados para entregas em hospitais, por exemplo.



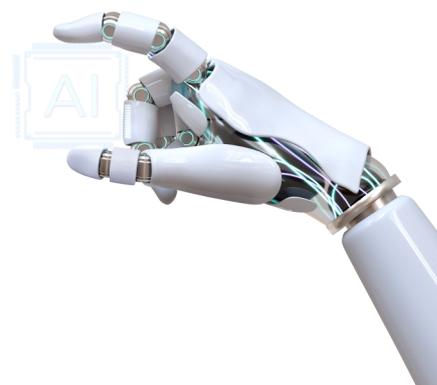
2) Viagens com o Hyperloop serão possíveis: Hyperloop é o nome dado para o **trem supersônico do Elon Musk**, que já está sendo produzido e testado nos Estados Unidos. Os inventores afirmam que no médio prazo poderemos ver este tipo de sistema funcionando para **transportar pessoas entre as cidades**.



3) Próteses darão aos humanos novas habilidades: a produção de **próteses** ficará tão boa e avançada em dez anos que elas poderão realmente dar **“novas habilidades” para as pessoas**. É possível que daqui a uma década, as próteses estarão tão avançadas que as pessoas irão se sentir bastante confortáveis em **unir tecnologia a seu próprio corpo**.

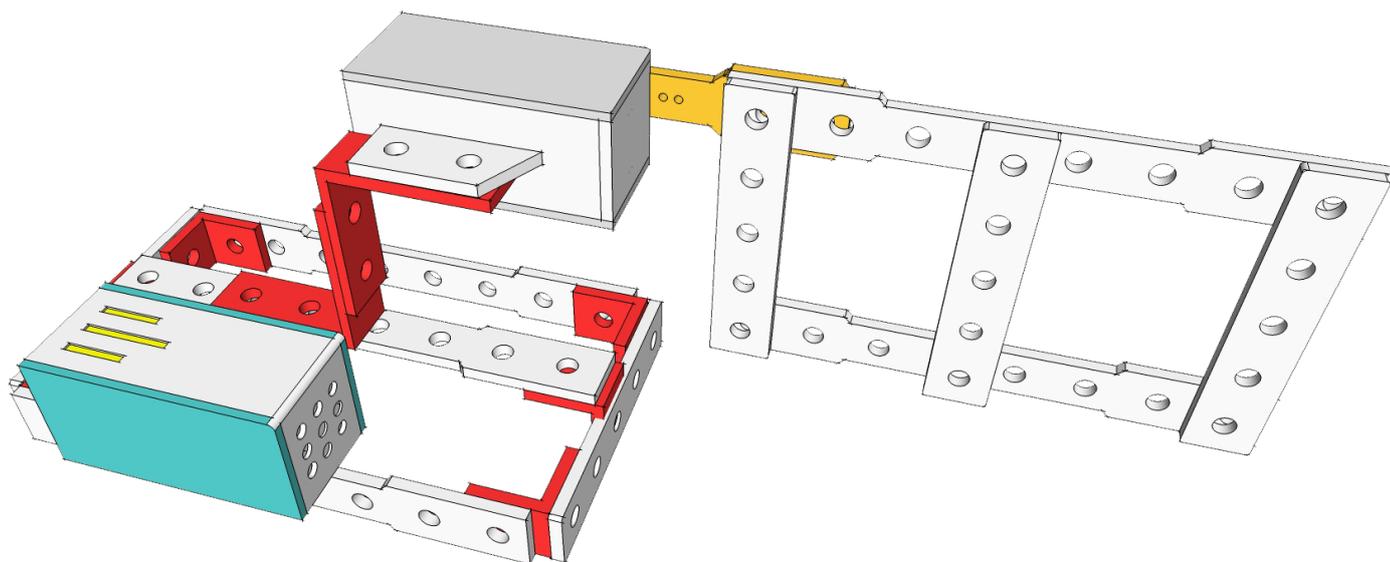
4) Roupas com poderes super-humanos: também em uma prospecção de 10 anos, os futuristas dizem conseguir ver que as **roupas poderão dar novas características às pessoas**, sugerindo que existirão calças do tipo leggings que tornarão os atos de correr ou andar mais fáceis. Outra sugestão é que um **traje parecido com o do homem aranha** seja criado por essa época, com polímeros em gel que **umentem a força**.

5) Robôs no serviço de casa: por volta de 2030 os robôs **poderão fazer parte do dia-a-dia de uma residência**. “Entre inteligência artificial e robótica, nós teremos muitos assistentes tecnológicos”. Estudiosos também apontam que estes **robôs funcionarão como companhia**, visto que muitas pessoas morarão sozinhas nessa época.



Montando um portão automático

A tecnologia é a aplicação da ciência. Para termos uma ideia mais clara a respeito dessa afirmação, que tal focarmos, por exemplo, nas **ondas sonoras**?! Elas são simplesmente um **fenômeno natural** que já vem sendo estudado pela **ciência chamada Física** há vários e vários anos. Se você estiver com seu **celular** agora, dentro do seu bolso, está com um belo exemplo de tecnologia, já que ele é sem dúvida **fruto dos diversos estudos que a ciência Física fez sobre as ondas sonoras**.



Outro exemplo seria um portão que se abre automaticamente, ou fecha, a partir de um sensor de som. Vejamos na prática!

Você vai precisar dos seguintes materiais: 5 barras longas 8 furos, 5 barras de 5 furos, 2 conectores L 4 furos, 6 conectores L 2 furos, 1 servo motor, 1 haste para servo S 2 furos, 1 sensor de som, placa Darwin, parafusos e porcas.

Para montar o portão faça o seguinte:

1. Monte a base com **3 barras longas 8 furos**, **2 barras de 5 furos** e **6 peças L de 2 furos**. Na parte central coloque o **conectore L 4 furos** e um outro ligado a ele, conforme o desenho.
2. Instale o **servo motor** na extremidade do segundo L de 4 furos, fixe a **haste de 2 furos no eixo do servo**. Depois disso, use mais **2 barras de 8 furos**, e **3 barras pequenas com 5 furos** para fazer o portão. Você pode usar papelão ou papel para montar um portão bem legal para sua estrutura.
3. Por fim acople o **sensor de som** em uma das laterais da base e então ligue-o na porta analógica **A0**. Conecte o servo motor **na porta D2**.

Programando o portão

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduíno**.

Como o **servo motor trabalha com ângulos de 0 a 180 graus**, pode ser que a posição inicial das hastes não reflita o movimento adequado de abrir e fechar o portão. Por esse motivo, **só fixe o portão nas hastes após ajustar os ângulos dos servos na programação abaixo**.

Programação:

```
#include <darwin6.h>

void setup() {
  ServoNaPorta(D2);
}

void loop()
{
  if ( LerSom(A0) == Sim ){
    MoverServo(D2, 90);
    EsperarSegundos(5);
  }
  else
  {
    MoverServo(D2, 0);
  }
}
```

Entendendo o que o programa faz:

Após coletar a leitura do **sensor de som conectado na porta A0**, perguntamos **SE existe a leitura de algum som nesta porta [if (LerSom(A0) == Sim)]**. Caso positivo (**Sim**) o servo vai se **posicionar em 90 graus** fazendo com que o portão se abra, e **esperar 05 segundos** para voltar à posição inicial. Caso o sensor não detecte algum som, o servo motor se **posicionará em 0**, fazendo com que o portão se feche.

Neste ponto talvez seja necessário você desparafusar a haste de acrílico do servo motor e fazer algumas experiências para descobrir onde fica a posição 0 e 90 em relação à sua montagem.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** do portão automático, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

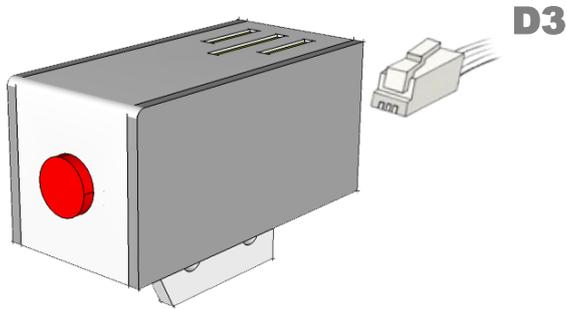
1) Das tecnologias que foram citadas nessa lição, qual delas chamou mais a sua atenção? Justifique.

2) Você acha que os robôs, além de fazer companhia e atividades domésticas, poderão fazer qualquer coisa que um ser humano faz? Explique.

3) Cite algumas tecnologias que ajudam pessoas com deficiência física? (Pesquise na Internet)

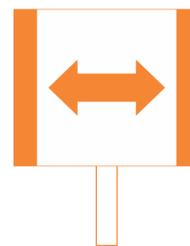
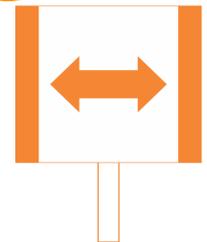
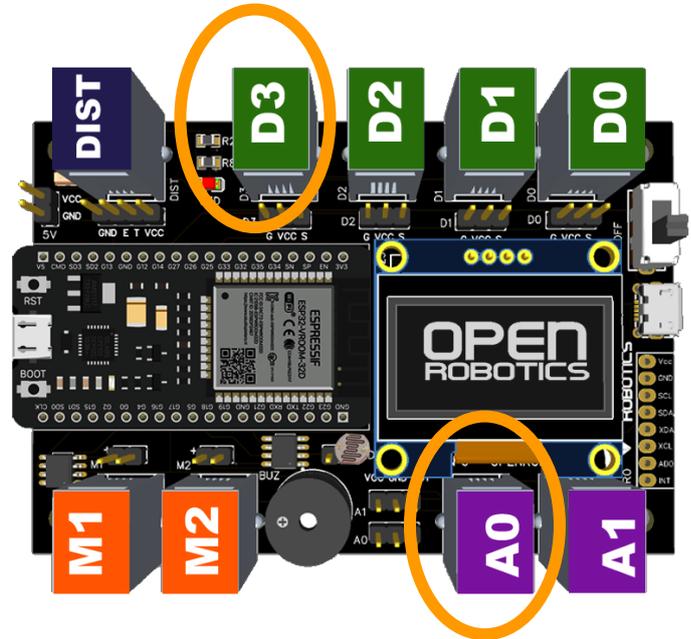
Sensores de inclinação

Os **sensores de inclinação** podem ser utilizados sempre quando você precisar detectar se algo está posicionado na **vertical** ou na **horizontal**. Eles são muito utilizados para testar se o **robô em movimento** tombou ou não. Vejamos como podemos utilizar o **sensor de inclinação**:



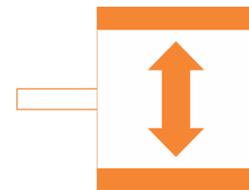
Programação:

```
#include <darwin6.h>
int Inclinao; //Variável para receber o
valor da inclinação.
void setup() {
  LedNaPorta(D3);
  AtivarComunicacao(); }
void loop()
{
  Inclinao = LerInclinao(A0);
  EnviarParaComputador(Inclinao);
  if (Inclinao == Vertical)
    LigarLed(D3);
  if (Inclinao == Horizontal)
    DesligarLed(D3); }
```



Horizontal = 1

Vertical = 0



Novidade na área:

Perceba que no programa acima utilizamos novamente as duas funções: **AtivarComunicacao()** e **EnviarParaComputador()**. Lembre-se de que com elas é possível receber e monitorar o valor sendo lido por um sensor em tempo real. Para isso, enquanto o programa estiver em execução, clique no **Ferramentas** do programa do Arduino e acione a opção **Monitor Serial**. Ao girar o sensor de inclinação, você vai perceber que **Horizontal equivale a 1** e **Vertical equivale a 0**.

Veículos robôs são uma realidade

Carros autônomos, também conhecidos como veículos autônomos ou self-driving cars, são veículos que utilizam tecnologias avançadas de robótica para operar de forma independente, sem a necessidade de intervenção humana constante. Esses veículos são projetados para perceber o ambiente ao seu redor, tomar decisões de direção e navegar com segurança em estradas e ambientes variados. Aqui estão alguns aspectos importantes relacionados a carros autônomos:

Níveis de Automação: A Sociedade de Engenheiros Automotivos (SAE) estabeleceu um sistema de níveis de automação, que varia de 0 (sem automação) a 5 (automação total). Atualmente, os carros autônomos mais avançados no mercado estão geralmente no nível 2 ou 3, o que significa que podem realizar algumas tarefas de direção de forma autônoma, mas ainda requerem supervisão humana.

Sensores e Percepção: Carros autônomos utilizam uma variedade de sensores, como câmeras, radares, lidar e sensores ultrassônicos, para perceber o ambiente ao seu redor. Esses sensores captam informações sobre obstáculos, sinalização de trânsito, pedestres, outros veículos e condições da estrada.

Sistemas de Navegação: Além de perceber o ambiente, carros autônomos dependem de mapas detalhados e sistemas de navegação para planejar rotas e entender a topografia do terreno.

Sistemas de Controle: A tomada de decisões é realizada por algoritmos de controle, muitas vezes baseados em aprendizado de máquina e inteligência artificial. Esses algoritmos interpretam as informações dos sensores para tomar decisões em tempo real, como mudanças de direção, aceleração e frenagem.

Desafios e Questões de Segurança: A implementação segura de carros autônomos enfrenta desafios significativos, incluindo a necessidade de lidar com situações imprevisíveis, tomar decisões éticas e garantir a segurança dos ocupantes e de outros usuários da estrada.

Testes e Desenvolvimento:

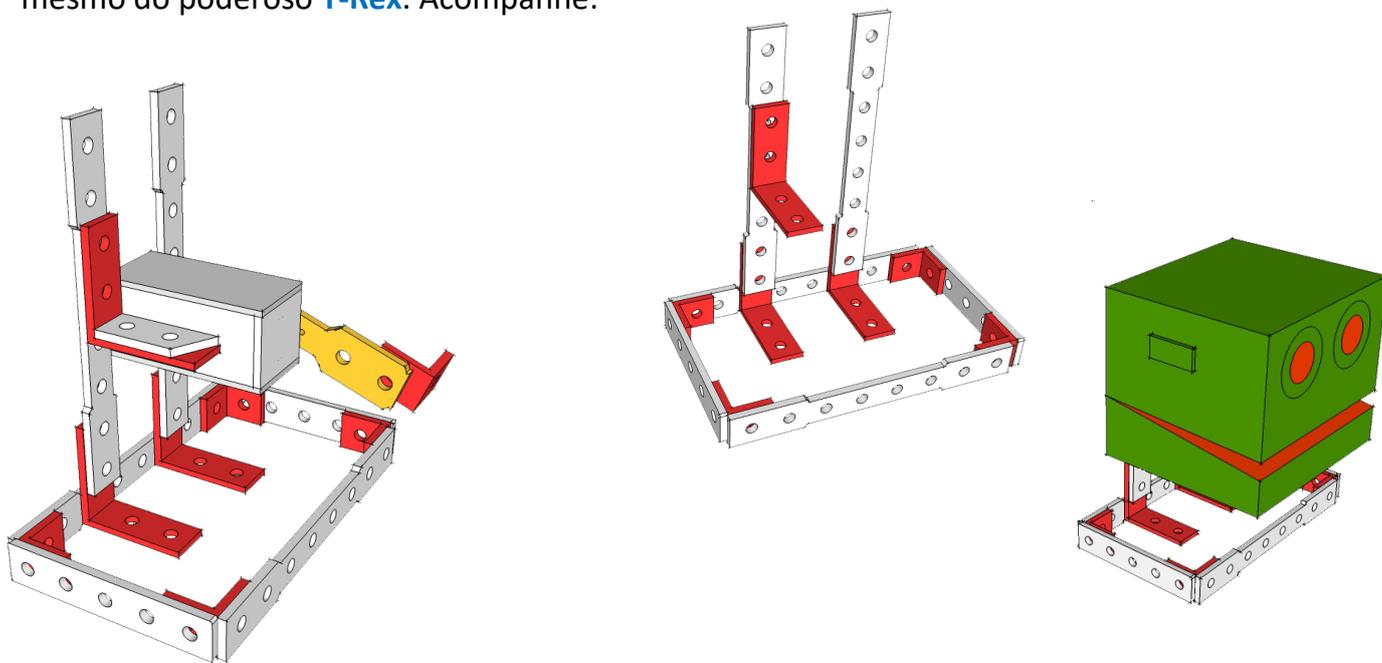
Muitas empresas, incluindo fabricantes de automóveis e empresas de tecnologia, estão investindo em pesquisas e testes extensivos para aprimorar a tecnologia de carros autônomos. Esses testes geralmente ocorrem em ambientes controlados e em estradas públicas para garantir a confiabilidade e segurança.



Montando uma cabeça de robô

A mordida de um tubarão branco da espécie *Carcharodon carcharias* pode chegar a **1,8 tonelada** - cerca de metade da força da mordida de um **Tiranossauro Rex**, segundo um estudo de pesquisadores australianos. Um exemplar dessa espécie com 2,4 metros e 240 kg tem, relativamente a suas dimensões e sua massa corporal, **uma das mordidas mais fortes do mundo**.

Vamos agora montar a nossa própria cabeça de robô para simular a mordida de um **tubarão** ou mesmo do poderoso **T-Rex**. Acompanhe:



Você vai precisar dos seguintes materiais:

1 servo motor, 1 haste para o servo S 2 furos, 1 sensor de inclinação, 4 barras longas 8 furos, 2 barras pequenas 5 furos, 3 conectores L 4 furos, 5 conectores L 2 furos, papelão cortado segundo os desenhos, placa Darwin, parafusos e porcas.

Para montar a cabeça de monstro faça o seguinte:

1. Comece montando a base com **2 barras longas de 8 furos, 2 barras de 5 furos e 4 peças L de 2 furos**. Em uma das laterais da base, acople os **2 conectores L 4 furos**.
2. Anexe mais duas barras longas de **8 furos aos conectores L 4 furos**. Depois instale o **servo motor** entre essas duas barras longas usando mais **1 peça L de 4 furos**. Fixe a **haste S 2 furos** no servo motor com o micro parafuso. Coloque o **conector L 2 furos** na extremidade da haste.
3. Monte a cabeça com o papelão envolvendo o servo motor de forma que a **mandíbula** possa ser fixada **na ponta do conector L 2 furos**. Use parafusos para fixar a mandíbula de modo que ela fique móvel.
4. Conecte o sensor de inclinação na **porta A0**, mas deixe-o livre na montagem para que você possa virá-lo à vontade. Isso porque a ideia é fazer o monstro abrir e fechar a boca em função da inclinação lida pelo sensor. O servo deve estar ligado na porta **D3**.

Programando a mandíbula do robô com o sensor de inclinação

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduino**. Na área de trabalho do seu computador dê dois cliques no ícone do **Arduino**. Então crie o programa mostrado abaixo:

Programação:

```
#include <darwin6.h>

int Inclinao;

void setup() {
  ServoNaPorta(D3);
}

void loop() {

  Inclinao = LerInclinao(A0);

  if ( Inclinao == Vertical )
    MoverServo(D3, 180);
  else
    MoverServo(D3, 90);
}
```

Entendendo o que o programa faz:

Primeiramente coletamos a leitura do **sensor de inclinação conectado na porta A0** e armazenamos na **variável Inclinao**.

Se o sensor de inclinação estiver na posição **Vertical**, o servo motor vai se posicionar no ângulo 180, simulando a abertura da boca do monstro. Ajuste esse ângulo na programação em função da posição da haste que você fixou. Exatamente por esse motivo, **é interessante você fazer testes primeiro antes de fixar a mandíbula na haste**.

Se o sensor de inclinação estiver na posição **Horizontal**, o servo motor vai se posicionar no ângulo 90, simulando o fechamento da boca do monstro. Novamente, pode ser necessário ajustar esse ângulo na programação em função da posição observada da haste.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** da cabeça do robô, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Pesquise na Internet sobre a existência do monstro do Lago Ness e anote aqui um breve resumo.

2) Seria possível dar movimento à boca de um esqueleto do T-Rex em um museu? O que você usaria?

3) Caso a gente resolvesse colocar os LEDs para piscarem nos olhos do monstro que montamos, seria necessário mudar os programas também? Por quê?

Robôs no fundo do mar

Um robô submarino, também conhecido como veículo submarino não tripulado (VANT) ou veículo autônomo subaquático (VAS), é uma máquina projetada para operar debaixo d'água sem intervenção humana direta. Esses robôs são utilizados em uma variedade de aplicações, desde pesquisas científicas e exploração oceânica até inspeções subaquáticas, monitoramento ambiental e missões de busca e resgate. Aqui estão alguns pontos relevantes sobre robôs submarinos:

Design e Estrutura: Robôs submarinos podem ter designs variados, mas geralmente incluem um invólucro hermético para proteger os componentes eletrônicos, sistemas de propulsão, sistemas de navegação, sensores e, em alguns casos, braços robóticos para realizar tarefas específicas.

Tipos de Propulsão: A propulsão de robôs submarinos pode ser realizada por meio de motores elétricos, hidráulicos ou outros métodos específicos para ambientes subaquáticos. Alguns modelos também podem imitar a propulsão de animais marinhos para maior eficiência.

Sensores e Instrumentação: Robôs submarinos estão equipados com uma variedade de sensores, como sonares, câmeras, sensores de salinidade, sensores de temperatura, entre outros, para coletar dados sobre o ambiente subaquático.

Aplicações Científicas: São usados em missões de pesquisa científica para explorar o fundo do mar, estudar ecossistemas marinhos, coletar amostras de água ou sedimentos, e realizar estudos geológicos e oceanográficos.

Inspeção e Manutenção Subaquática: Robôs submarinos são empregados em operações de inspeção de estruturas subaquáticas, como plataformas de petróleo, cabos submarinos, dutos, cascos de navios e estruturas submersas.

Exploração de Áreas Inacessíveis: Permitem a exploração de áreas inexploradas ou inacessíveis para mergulhadores, como regiões profundas do oceano ou áreas com condições perigosas.

Busca e Resgate: Em situações de emergência, como acidentes de avião ou naufrágios, robôs submarinos podem ser utilizados para realizar buscas e operações de resgate.

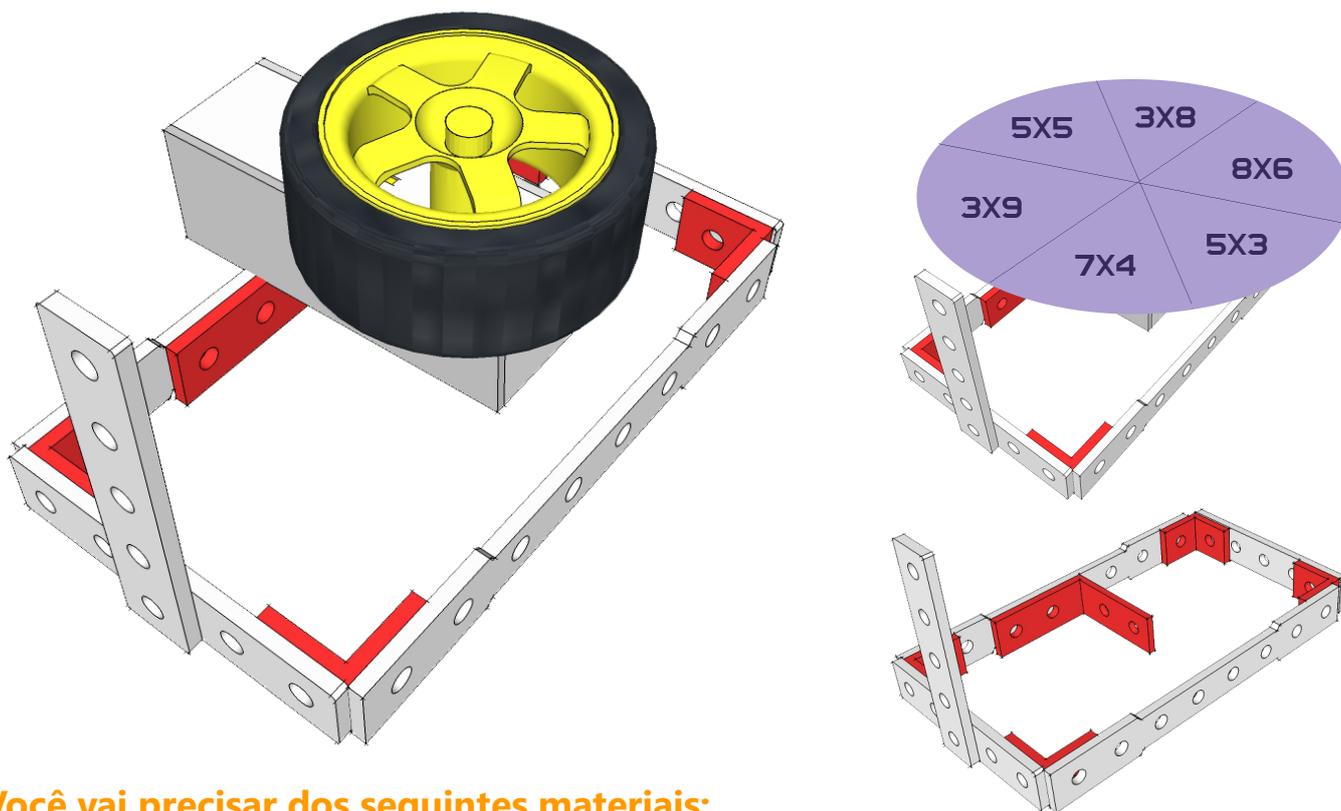
Operação Remota e Autônoma: Podem ser controlados remotamente por operadores humanos, utilizando comunicação via cabo, acústica ou wireless, ou podem operar autonomamente, seguindo rotas predefinidas ou respondendo a comandos baseados em algoritmos de inteligência artificial.



Criando o jogo da roleta

Se a gente tivesse uma **roleta com alguns números marcados**, montada sobre uma estrutura que permitisse girá-la para que ela parasse **selecionando um dos números aleatoriamente** a cada acionamento, daria para **brincar de Matemática de algumas formas diferentes**. Adivinhar qual número será escolhido já seria uma brincadeira interessante.

Descobrir qual membro da equipe consegue multiplicar mentalmente mais rápido um primeiro número sorteado pelo segundo é outro bom exemplo. O mesmo vale para a subtração e a soma. A seguir você verá como montar essa estrutura. Acompanhe:



Você vai precisar dos seguintes materiais:

1 motor CC, 1 roda de borracha, 1 botão de toque, 2 barras 8 furos, 3 barras de 5 furos, 1 conector L 4 furos, 4 conectores L com 2 furos; 1 roleta feita em papelão, placa Darwin, parafusos e porcas.

Para montar o jogo da roleta faça o seguinte:

1. Comece montando a base com **as barras de 8 e 5 furos conectadas por 4 peças L de 2 furos**. Depois utilize um a barra de 5 furos para fazer o poste de marcação do sorteio. Ela será o **marcador** necessário para selecionar um número qualquer na roleta.
2. Na lateral oposta fixe outro **conector L 4 furos** e então acople o **motor CC** nele. Depois disso, anexe a **roda** ao eixo do motor CC que estiver para cima. Usando uma **fita adesiva fixe a roleta de papelão sobre a roda**.
3. O **botão de toque** pode ficar solto na montagem. Ele deve ser conectado **na porta D3**.
4. O motor CC deve ser conectado **na porta M1**. Como vamos utilizar o motor CC, também é necessário conectar **a fonte de alimentação (+ -) na placa Darwin**.

Programando nosso jogo da roleta

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduino**.

Programação:

```
#include <darwin6.h>

void setup() {
  BotaoNaPorta(D3);
  AtivarMotores();
}

void loop() {
  if ( LerBotao(D3) == Sim )
  {
    MotorM1(Frente);
    EsperarSegundos ( SortearNumero(1, 5) );
    MotorM1(Parar);
  }
}
```

Entendendo o que o programa faz:

O segredo nesse programa é fazer com que **o motor CC gire a roda** (e conseqüentemente a roleta de números) assim que a **pessoa pressionar o botão**.

Primeiramente coletamos a leitura do **sensor de toque (botão) na porta D3**.

No próximo passo perguntamos **SE** o botão está pressionado. Caso esteja (**Sim**), vai começar a girar o motor CC por um **tempo sorteado entre 01 e 05 segundos**, para que não haja manipulação externa. Por fim, **paramos o motor CC** para vermos qual número da roleta ficou perto do **marcador**.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** do jogo da roleta, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Quando você atravessa uma rua, está usando a matemática? Explique.

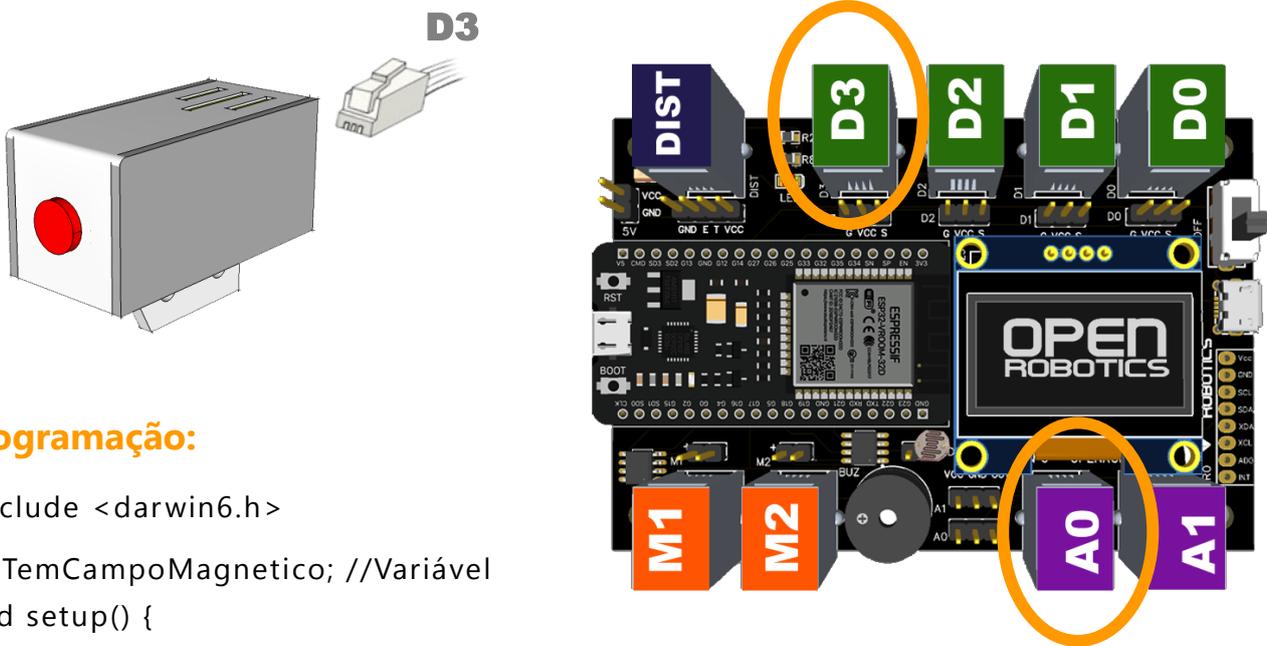
2) Para você, quais são os principais benefícios em ter habilidades na matemática?

3) Ter uma vida financeira saudável depende de ser amigo da matemática? Dê sua opinião.

Utilizando o sensor magnético

Os **sensores magnéticos** devem ser usados quando você precisa detectar um **campo magnético**, como por exemplo, um **ímã**. Eles podem ser muito úteis quando você precisar detectar algo sem ter que encostar neste objeto, uma vez que o ímã gera um **campo ao seu redor**, invisível, mas que pode ser **detectado com este sensor**. Para essa montagem utilize **um ímã**, que deve ser **posto em contato com a frente do sensor durante os testes**.

Ligando um LED quando há campo magnético



Programação:

```
#include <darwin6.h>
int TemCampoMagnetico; //Variável
void setup() {
  LedNaPorta(D3);
  AtivarComunicacao();
}
void loop()
{
  TemCampoMagnetico = LerMagnetico(A0);
  EnviarParaComputador(TemCampoMagnetico);
  if (TemCampoMagnetico == Sim)
    LigarLed(D3);
  else
    DesligarLed(D3);
}
```

Desafios desta aula:

1. Crie uma cancela e faça com que ela se abra ou feche conforme a aproximação do ímã.
2. Faça a sirene interna da placa Darwin apitar 3 vezes quando o ímã aproximar.

A ciência dos Robôs

Robótica é a ciência e técnica da concepção, construção e utilização de robôs. Ela é também um ramo educacional e tecnológico que engloba computadores, robôs e computação, que trata de sistemas compostos por partes **mecânicas automáticas e controladas por circuitos integrados**, tornando sistemas mecânicos motorizados, controlados manualmente ou automaticamente por circuitos elétricos. As máquinas, pode-se dizer que são vivas, mas ao mesmo tempo são uma imitação de algumas habilidades humanas ou de animas; feitos de fios unidos e mecanismos, isso tudo junto concebe um robô. Cada vez mais as pessoas utilizam os robôs para suas tarefas. Em um futuro os robôs possivelmente serão extremamente populares. Esta tecnologia, hoje adaptada por muitas fábricas e indústrias, tem obtido de um modo geral, êxito em questões levantadas sobre a redução de custos, aumento de produtividade.

A ideia de se construir robôs começou a tomar força no início do século XX com a necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. É nesta época que o robô industrial encontrou suas primeiras aplicações, o pai da robótica industrial foi **George Devol**. Devido aos inúmeros recursos que os sistemas de microcomputadores nos oferecem, a robótica atravessa uma época de contínuo crescimento que permitiu, em um curto espaço de tempo, o desenvolvimento de robôs inteligentes.

A robótica também faz parte do imaginário dos amantes da ficção científica. Nesta área o termo Robótica foi popularizado pelo escritor de Ficção Científica Isaac Asimov, na sua ficção "I, Robot" (Eu, Robô), de 1950. Neste mesmo livro, Asimov criou leis, que segundo ele, regeriam os robôs no futuro.

Leis da robótica na ficção científica:

- 1) Um robô não pode ferir um ser humano ou, por ócio, permitir que um ser humano sofra algum mal.
- 2) Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.
- 3) Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira e Segunda Leis.

No mundo real o homem sempre busca nos robôs a construção de máquinas que possam resolver problemas que seriam perigosos para o ser humano, por exemplo, criando robôs que podem salvar vidas em ambientes em que pessoas estejam correndo perigo e precisem ser resgatadas, de um desastre natural, por exemplo.

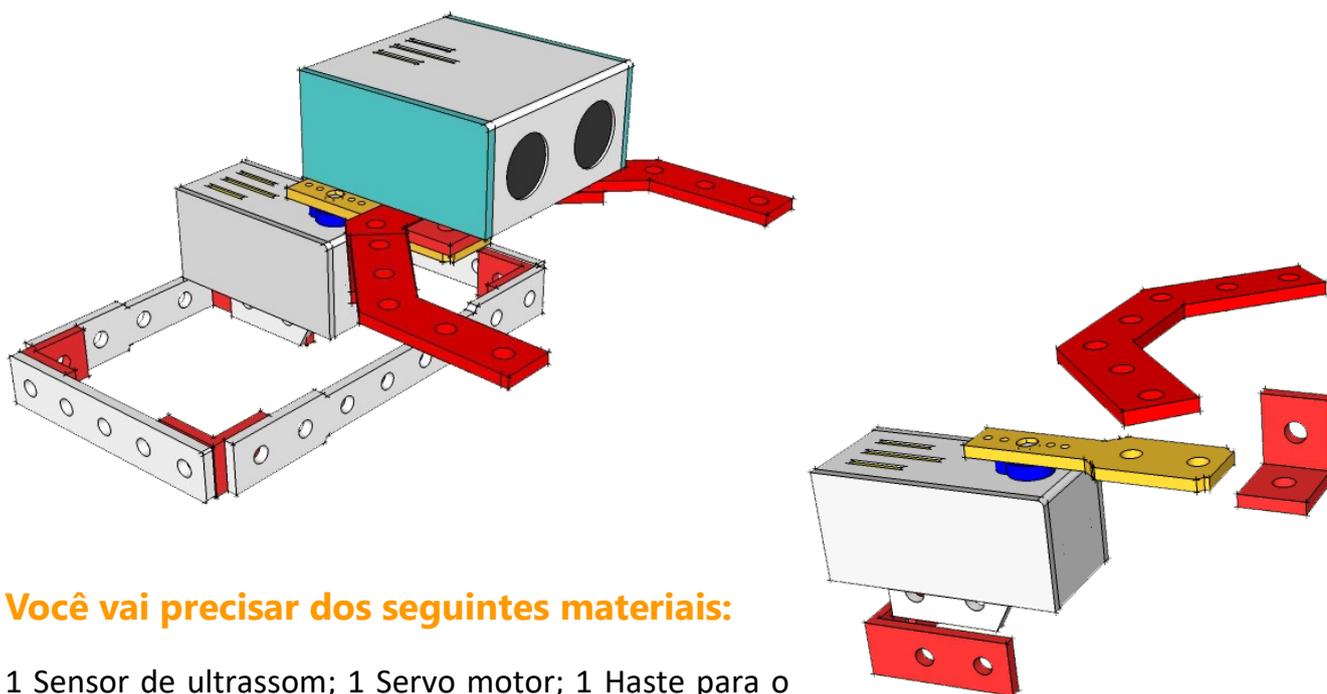


Montando um robô que busca por objetos

É muito bacana quando a gente pega uma bolinha para brincar com nosso cão e o bichinho fica louco olhando de um lado e de outro tentando acompanhar o movimento, mesmo estando a tão desejada esfera ainda em nossas mãos.

Depois da Robótica apareceu também a tal **Inteligência Artificial**. No meio das discussões entre o que é, e o que não é, possível transferir de intelecto para uma máquina, um grupo de pessoas gosta de dizer que os **robôs e máquinas podem se comportar de forma inteligente**.

No seu próximo desafio você construíra um belo exemplo de “comportamento inteligente”, vamos criar uma cabeça de robô que procura por objetos próximos.



Você vai precisar dos seguintes materiais:

1 Sensor de ultrassom; 1 Servo motor; 1 Haste para o servo motor; 2 Barras 8 furos; 2 barras de 5 furos; 1 Conector L 7 furos; 2 conectores V com 5 furos; 1 Conector L 4 furos; 5 Conectores L 2 furos; Placa Darwin, Parafusos e porcas.

Para montar o robô faça o seguinte:

1. Comece montando a base com **2 barras de 5 furos e 2 barras longas de 8 furos conectadas por 4 peças L de 2 furos**. Na peça central de uma delas, fixe a peça **L com 4 furos** e nela conecte o servo motor. Ligue no servo motor a Haste pequena com 2 furos, na ponta desta haste conecte uma **peça em L pequena com 2 furos**, e nela fixe o sensor de distância por ultrassom. Para os braços do robô utilize **1 conector L 7 furos** e mais **2 conectores V com 5 furos**.
2. O servo motor deverá ficar ligado **na porta D3** da placa do Inventor.
3. Já o sensor de distância deverá ser colocado **na porta DIST**.

Programando nosso robô que procura objetos

Após a realização da **montagem** vamos programar a nossa estrutura e **enviar o código** para a CPU do **Inventor**. Para isto você vai utilizar o código do **Arduino**.

Programação:

```
#include <darwin6.h>
int Distancia; int Angulo;
void setup() {
  ServoNaPorta (D3);
}
void loop()
{
  Distancia = LerDistancia ();
  if (Distancia > 10) {
    Angulo = SortearNumero (50, 180);
    MoverServo (D3, Angulo);
    EsperarSegundos (0.5);
  }
}
```

Entendendo o que o programa faz:

Primeiramente são definidos dois locais na memória para armazenar dados (**variáveis**):

Uma variável para armazenar a **distância** do objeto que será colocado na frente do robô;

E a outra para armazenar um **ângulo** que será sorteado. Depois informamos que vamos usar um servo motor **na porta D3**.

Em seguida fazemos a **leitura da distância** do objeto utilizando o sensor de ultrassom. **Se esta distância for maior do que 10 centímetros**, então o robô ficará procurando um objeto movendo a cabeça para um lado e para o outro. Isso é feito através do **sorteio de um número que é enviado como ângulo para o servo motor**.

Perceba que você pode mudar os valores para o sorteio, mas eles **deverão variar entre 0 e 180**, pois estes são os ângulos possíveis para o servo motor.

Ampliando seus conhecimentos

Após a realização da **montagem e programação** do robô, vamos **agora avaliar o conhecimento que adquirimos**. Para isso responda individualmente ou em equipe as questões abaixo:

1) Pesquise na Internet como os morcegos fazem para desviar de objetos. Você vai ter uma surpresa.

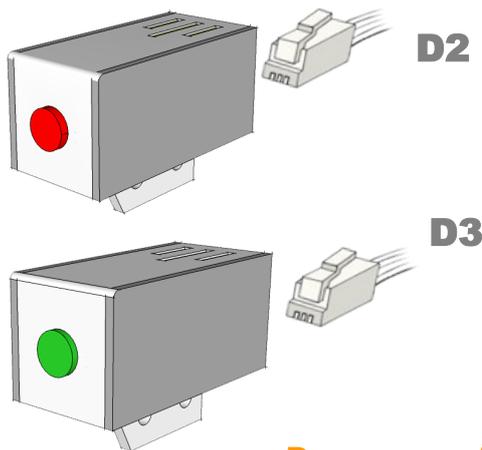
2) Atualmente alguns veículos possuem sensores de estacionamento, você sabe como eles funcionam?

3) Quais outros usos poderíamos dar para os sensores de distância?

Enviando comandos do computador para o robô

Em várias situações é necessário **enviar comandos do computador para o robô**, para que ele realize alguma atividade. Para isso podemos utilizar o **monitor serial**. Com ele podemos **digitar algum valor** no teclado, que ele será **lido pelo robô** e executado conforme a programação dada.

Acionando LEDs pelo computador



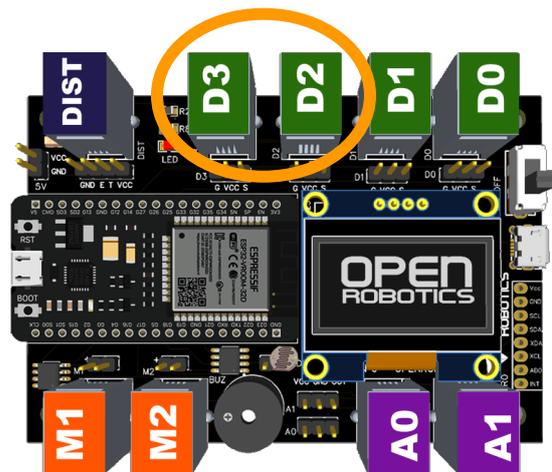
Programação:

```
#include <darwin6.h>
int comando; //Variável para receber o valor
computador.
void setup() {
  LedNaPorta(D2); LedNaPorta(D3); AtivarComunicacao();
}
void loop() {
  comando = ReceberNumeroPeloComputador();
  if (comando == 1)
    LigarLed(D2);
  if (comando == 2)
    LigarLed(D3); }
```

Depois de carregar o programa abra o **Monitor serial** no menu **Ferramentas**. Então digite **nesta caixa os comandos 1 ou 2** e clique no botão **Enviar**.

Desafios desta aula:

1. Altere o programa para que você possa também desligar os LEDs com comandos do computador.
2. Acrescente uma sirene que também deverá ligar ou desligar.

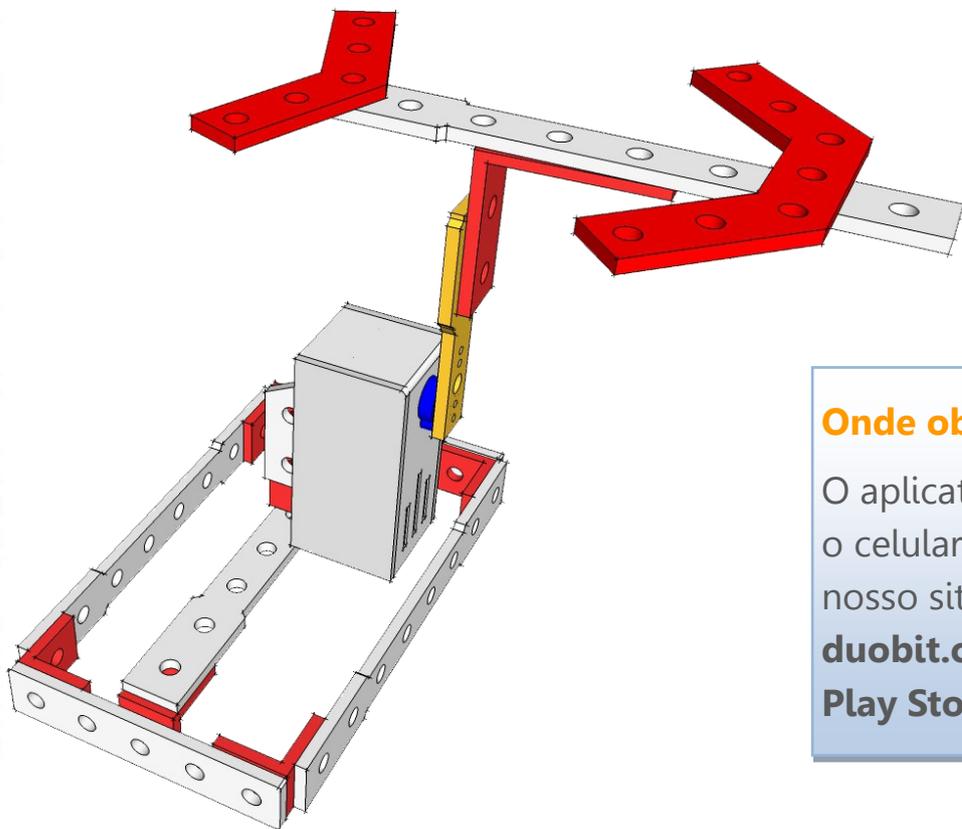


do



Controlando um avião por bluetooth

Vamos agora fazer uma montagem que precisa de um controle apropriado para pilotar aviões, embarcações e outros dirigíveis, **mais conhecido como manche**. A ideia é fazer com que a inclinação do celular controle a **inclinação da estrutura montada**, fazendo dele um **manche**.



Onde obter o aplicativo?

O aplicativo **OpenController** para o celular pode ser baixado em nosso site

duobit.com.br/downloads ou na **Play Store**.

Para esta montagem você vai precisar dos seguintes materiais:

1 servo motor; 1 haste para o servo S 2 furos; 4 barras 8 furos; 2 barras 5 furos; 1 conector L 7 furos; 1 conector V com 5 furos; 2 conectores L 4 furos; 6 conectores L 2 furos; 1 celular com o **aplicativo OpenController instalado**; placa Darwin, parafusos e porcas.

Agora vamos montar nossa estrutura:

Comece montando a base com **3 barras de 8 furos e 2 de 5 furos conectadas por 6 peças L de 2 furos**. Na peça central, coloque **uma peça L 4 furos**, para conectar o servo motor.

Instale o **servo motor** na extremidade da **peça L 4 furos** e então conecte a **haste S 2 furos** com o parafuso apropriado. Agora fixe o **conector V 5 furos** na extremidade da **barra 8 furos** que sobrou. Depois, na outra extremidade da barra, pule um furo e fixe o **conector L 7 furos** de forma a ficar centralizado na barra. Falta ainda conectar esta última etapa montada à **haste S 2 furos** usando a **peça L de 4 furos que sobrou**. Não se esqueça de ligar o servo motor na porta **D3**.

Programando nosso avião

Agora vamos ver como podemos utilizar o celular para controlar o avião. Veja que você vai precisar do nosso aplicativo **OpenController**. Uma vez o aplicativo instalado, é necessário **parear o celular com o bluetooth** presente na placa Darwin, para depois abrir o aplicativo e então **fazer a conexão**. Além disso, a placa Darwin deve estar carregada com o **código Arduino** abaixo:

Programação:

```
#include <darwin6.h>

int Inclinacao;

void setup() {
  AtivarBluetooth("DARWIN");
  ServoNaPorta(D3);
}

void loop()
{
  if (BluetoothEstaRecebendo() == Sim)
    Inclinacao = ReceberNumeroPeloBluetooth();

  if (Inclinacao >= 0 )
    MoverServo(D3, Inclinacao);
}
```

Atenção:

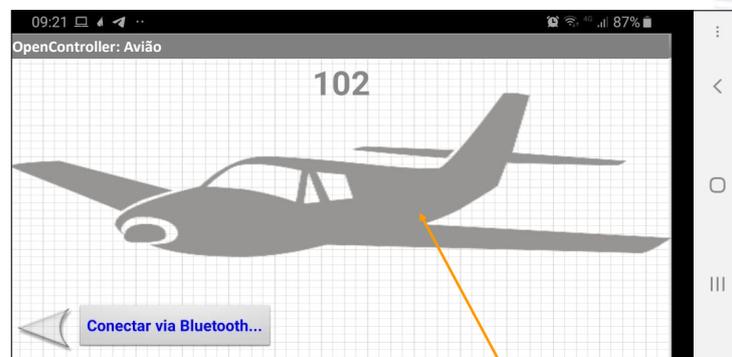
Antes de utilizar o aplicativo, não se esqueça de parear o seu celular com a placa Darwin.

Entendendo o que o programa faz:

O algoritmo do **Arduino** recebe a inclinação vinda do sensor acelerômetro do celular e a repassa para a inclinação do servo motor. Então você deve navegar no aplicativo até chegar **nessa tela** para depois começar a inclinar o celular de um lado para o outro para ver o avião refletir o movimento.

No programa **Arduino**, primeiramente criamos uma variável para receber a **inclinação** que virá via Bluetooth do celular ou tablet. Na sequência fazemos as configurações iniciais para receber os dados do Bluetooth, além de declarar em qual porta está o servo motor.

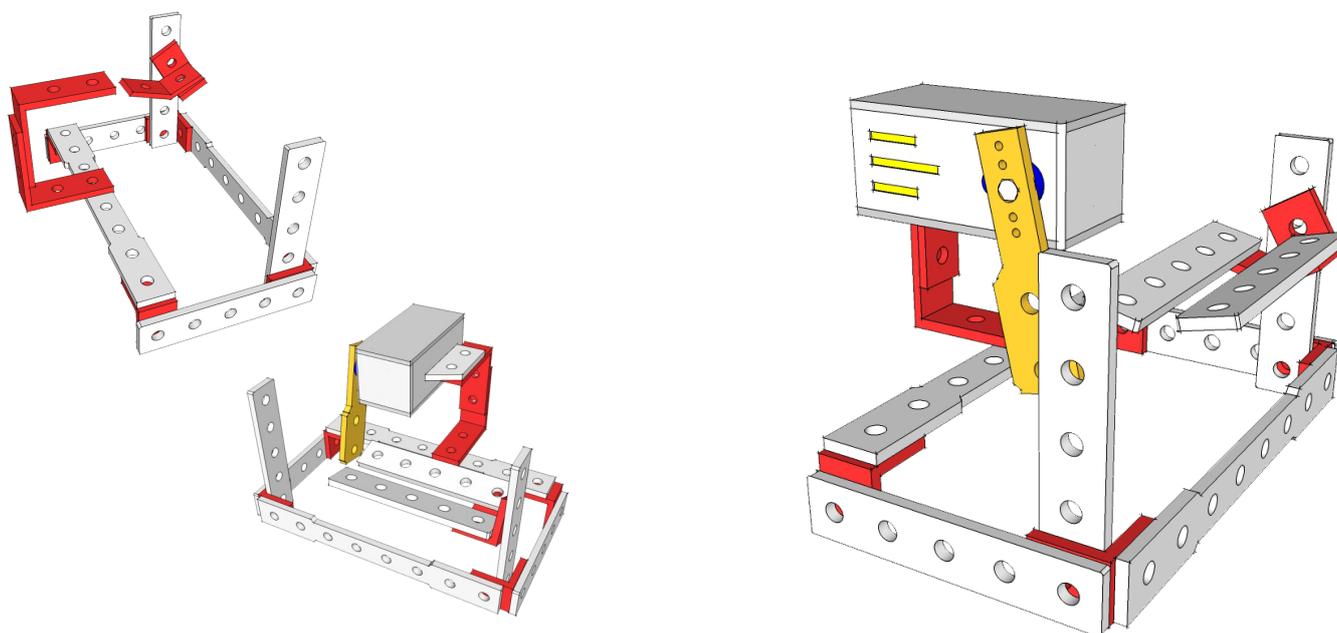
Se o sinal de Bluetooth estiver disponível, então converte o valor recebido em um número por meio da função **ReceberNumeroPeloBluetooth()**. Uma inclinação igual ou maior do que zero grau pode ser passada para o servo motor, posicionando o seu eixo exatamente na inclinação recebida do celular ou tablet.



Simulando uma catapulta medieval

Uma máquina que lança bolinhas

Quer ficar bom em alguma coisa? Treine, treine, treine, treine, treine, treine, treine e depois, treine... A **evolução tecnológica** trouxe também algumas máquinas que ajudam as pessoas a treinarem. Hoje já tem diversas aplicações e uma das mais interessantes é a máquina que ajuda os goleiros dos times de futebol a treinar. Tem também aquela que **atira as bolinhas** para os tenistas poderem rebater. Na próxima montagem você vai criar uma maquinha dessas para ver como é simples o princípio delas.



Para esta montagem você vai precisar dos seguintes materiais:

1 Botão de toque; 1 Servo motor; 2 Barras 8 furos; 6 Barras 5 furos; 1 Haste para servo; 2 peças L 4 furos e 5 L 2 furos; 1 peça V pequena 45 graus; placa Darwin; bolinha de ping-pong, parafusos e porcas

Agora vamos montar nossa estrutura:

Comece montando a base com **2 barras de 8 furos e 2 de 5 furos conectadas por 4 peças L de 2 furos, conforme o desenho**. Na parte de traz da base, coloque **2 peças L 4 furos**. Uma vai segurar o servo motor, a outra estará presa à base. No servo motor parafuse a haste de 2 furos.

Na parte da frente da base, suba 2 barras de 5 furos, uma em cada extremidade.

Para fazer a rampa onde a bola ficará esperando para ser lançada, utilize **1 peça em V 45 graus com 2 furos e 1 peça L de 2 furos**. Nelas conecte as **2 barras de 5 furos** para formar o caminho para a bola percorrer até ser lançada.

Conecte o **servo na porta D3** e o **botão de toque na porta D2**.

Programando a catapulta

Agora vamos programar a nossa catapulta. Depois de fazer a programação, experimente colocar energia extra na placa. Você vai perceber que isso traz muito mais força para o lançamento.

Programação:

```
#include <darwin6.h>

int Botao;

void setup(){
    ServoNaPorta(D3);
    BotaoNaPorta(D2);
}

void loop(){
    Botao = LerBotao(D2);
    if (Botao == Sim)
        MoverServo(D3, 180);
    if (Botao == Nao)
        MoverServo(D3, 90);
}
```

Entendendo o que o programa faz:

No código em **Arduino** a programação começa com a importação da biblioteca **<darwin.h>** que contém a programação que deixa a **sintaxe dos comandos** muito mais fácil.

Depois vem a definição da **variável Botao** que vai armazenar a leitura feita pelo botão de verdade.

Então é realizada a leitura do botão que está **na porta D2** e o resultado é colocado na **variável Botao**.

Na sequência existe um **teste para verificar se o botão está pressionado**, em caso afirmativo (**Sim**) então o servo que está **na porta D3** da placa irá se movimentar.

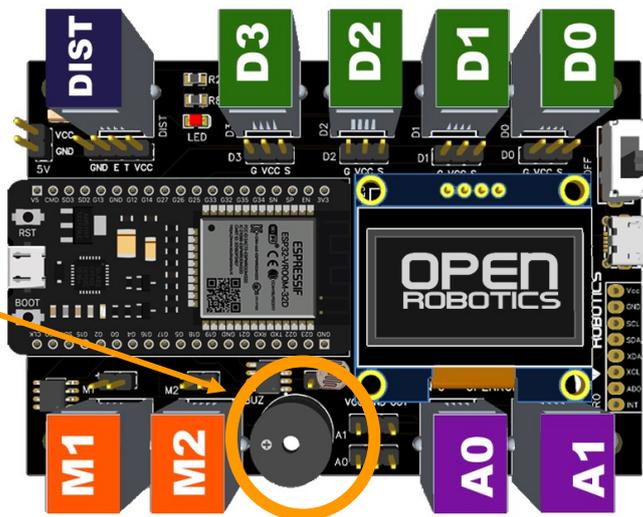
Veja que talvez você tenha que alterar os ângulos, dependendo de como ficou a posição do servo que você instalou na placa.

Acionando a sirene interna

A placa Darwin possui também uma **sirene interna** para permitir que se possa ligar e desligar um aviso sonoro vindo da própria placa.

A **sirene interna** da placa Darwin está na **porta D23**. Saber isso é importante para conseguir acessá-la e usar nas montagens de robótica. Mas para facilitar, criamos uma variável chamada `SireneInterna` que você pode utilizar para acessá-la.

Sirene interna



O programa abaixo é um exemplo de uso da sirene interna. Digite-o e **carregue** na placa para experimentar.

Programação:

```
#include <darwin6.h>

void setup()
{
  //Informando que sirene interna está na porta digital D23 denominada também SireneInterna
  SireneNaPorta ( SireneInterna );
}

void loop()
{
  LigarSirene ( SireneInterna );
  EsperarSegundos ( 2 );
  DesligarSirene ( SireneInterna );
  EsperarSegundos ( 2 );
}
```

Utilizando o Display OLED

O display OLED interno na placa Darwin facilita (e muito) nosso trabalho de visualização de dados e informações saindo ou chegando na nossa [placa Darwin](#).

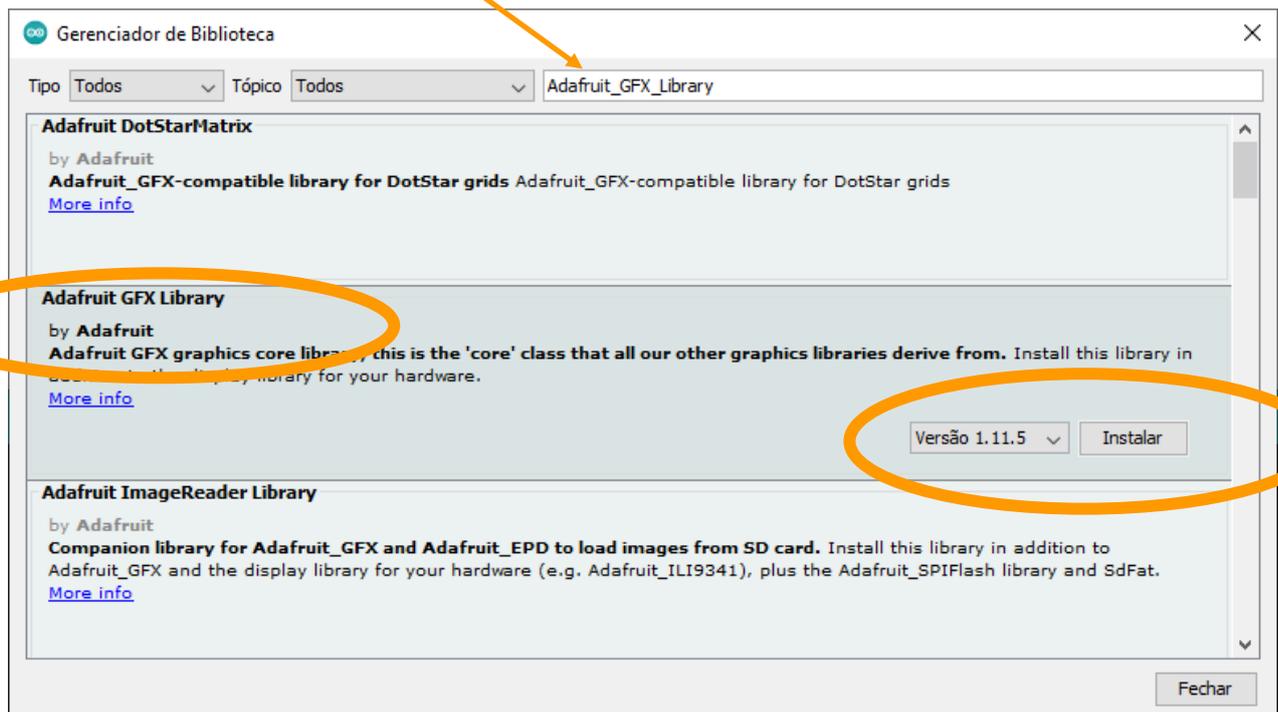
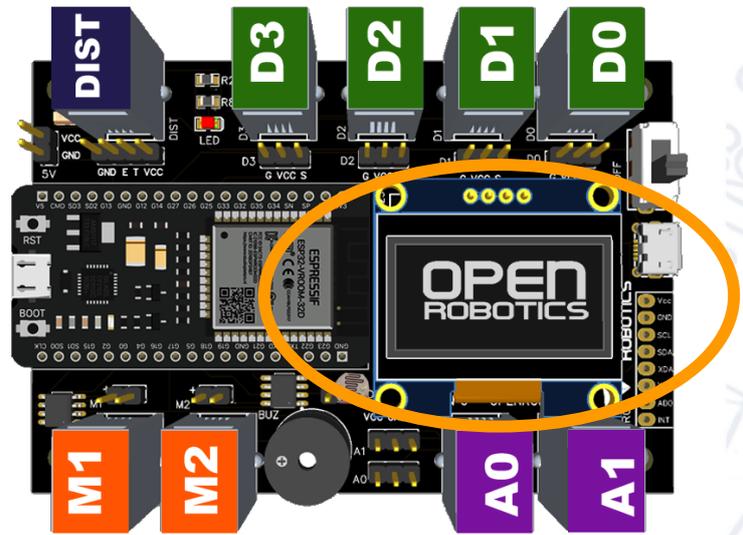
Ele possui uma biblioteca com dezenas de recursos diferentes, que vão desde a [visualização de textos](#), até a criação de animações.

Neste momento vamos utilizar o [display](#) para exibir alguns textos, variando fontes, tamanhos e espaçamentos.

Instalando o driver do Display OLED

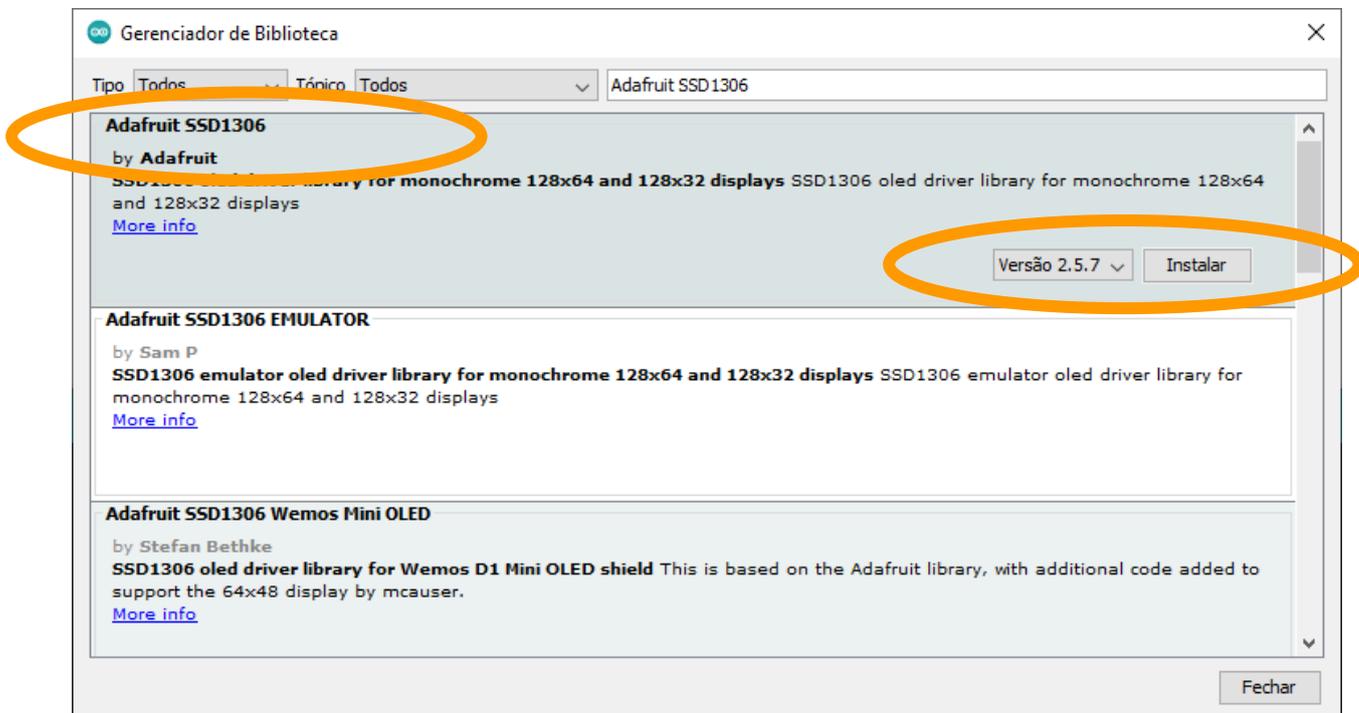
Antes de utilizar o display, precisamos instalar os drivers que vão fazer com que ele funcione. Siga o passo a passo abaixo, vai ser bem fácil.

- Clique no menu [Ferramentas, Gerenciar bibliotecas](#).
- Digite [Adafruit_GFX_Library](#) e clique em [Instalar](#)



Agora vamos instalar um segundo driver necessário.

- Clique novamente no menu **Ferramentas, Gerenciar bibliotecas.**
- Digite **Adafruit_SSD1306** e clique em **Instalar**



Agora vamos ver como podemos acionar o nosso display.

Programação:

```
#include <darwin6.h>
#include <darwin6_display.h>
void setup()
{
  AtivarDisplay();
  EsperarSegundos(5);
  LimparDisplay();
}
void loop()
{
  EscreverDisplay(0,0,1,"TAMANHO 1");
  EscreverDisplay(0,10,2,"TAMANHO 2");
  EscreverDisplay(0,30,3,123456);
}
```

Como funciona:

Após concluir a digitação, envie o programa para a placa Darwin.

A função `AtivarDisplay()`, ativa o display para que você possa utiliza-lo na sequencia. Quando o display é ativado, automaticamente uma mensagem da Duobit aparece por alguns segundos.

Na sequencia do código limpamos o display.

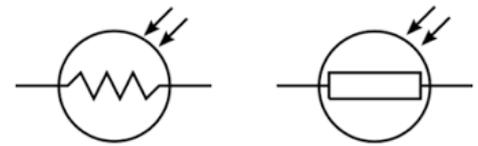
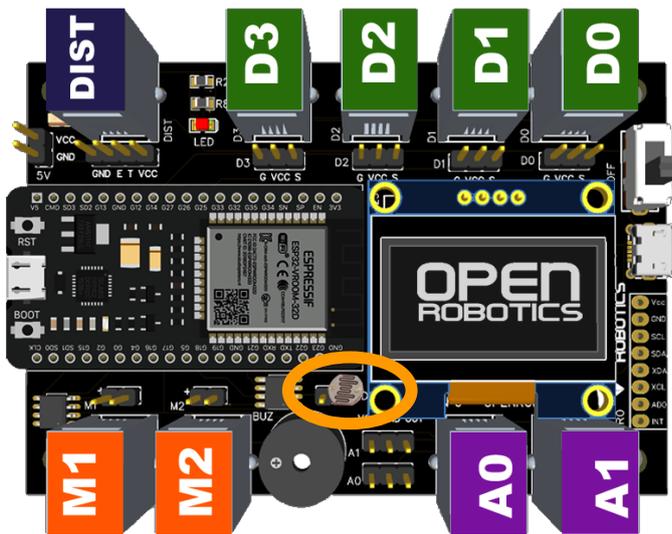
O comando para escrever no display possui a seguinte sintaxe:

EscreverDisplay(posX, posY, TamanhoDaFonte, TextoOuNumero)

Utilize 1,2 ou 3 para o tamanho da fonte. Para escrever números não é necessário aspas. Para escrever textos, é necessário a colocação das aspas.

Sensor de Luz interna

Agora vamos ver como podemos medir a **intensidade da luz** no ambiente onde estamos. Para isso vamos utilizar um **sensor de luz**, também chamado de **LDR (Light Dependent Resistor)**. Com ele vamos ativar um **LED** quando estiver escuro e apagaremos o LED quando estiver claro, **simulando** o que acontece com o poste de luz nas ruas.



Programação:

```
#include <darwin6.h>
#include <darwin6_display.h>

void setup()
{
  AtivarDisplay();
  EsperarSegundos(5);
  LimparDisplay();
}

void loop()
{
  if ( LerLuzInterna() == Escuro )
    EscreverDisplay(0,0,2,"Escuro");
  else
    EscreverDisplay(0,0,2,"Claro ");
  EsperarSegundos(0.1);
  LimparDisplay();
}
```

Entendendo o código:

Para que funcione o código ao lado, vamos precisar do display instalado, conforme você já viu anteriormente.

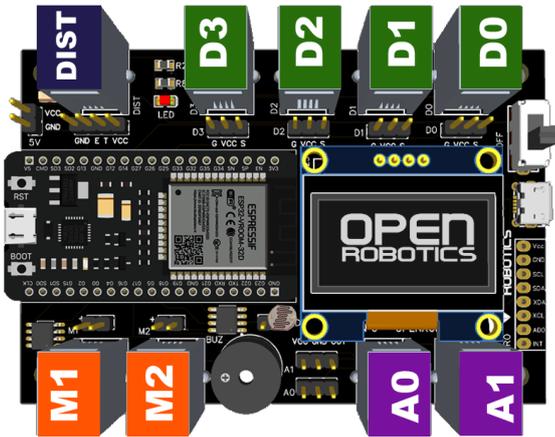
A função que realiza a medição da quantidade de luz é chamada **LerLuzInterna()** e ela retorna 0 (Escuro) ou 1 (Claro).

Após digitar o código abaixo, envie para a placa Darwin e passe a mão sobre o sensor de luz.

No display irá aparecer a palavra Claro ou Escuro conforme a condição da iluminação.

Conectando na rede Wifi

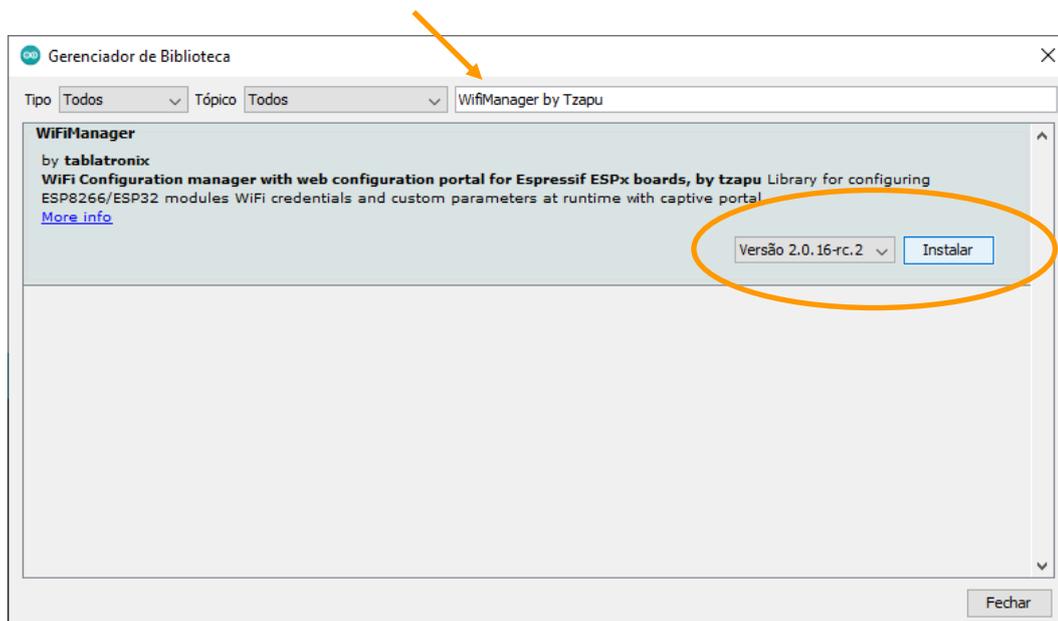
Agora vamos conhecer como podemos conectar a nossa placa **Darwin** em uma rede **Wifi**. Após realizar essa etapa, sua placa já pode se conectar em milhares de recursos espalhados pela Internet, por exemplo: conexão com **Telegram**; Conexão com **banco de dados**; Conexão com a **Alexa**; conexão com o **WhatsApp**, e diversos outros recursos web.



Para que possamos estabelecer uma comunicação entre a placa Darwin e a rede Wifi, primeiramente precisamos instalar a biblioteca padrão de comunicação, a qual vai permitir estabelecer uma **conexão rápida e segura** com um roteador.

Passo 1. Instalando a biblioteca WifiManager by Tzapu

Para instalar a biblioteca, utilize o **Gerenciador de Bibliotecas** dentro do menu **Ferramentas**, **Gerenciar Bibliotecas**. Escolha a biblioteca **WifiManager by Tzapu** e clique em Instalar.



Passo2: Digite o programa abaixo no Arduino e envie para a placa

Para que a placa **Darwin** possa reconhecer a sua rede, é preciso enviar a programação abaixo. É um código muito simples, que transforma a placa Darwin em um **ponto de acesso**, ou seja, após ela executar o código abaixo, já podemos nos conectar no Placa Darwin pelo celular, para que possamos informar os dados da nossa **rede local**.

Programação:

```
#include <darwin6.h>
#include <darwin6_display.h>
#include <darwin6_wifi.h>

void setup()
{
  AtivarWifi();
}

void loop()
{
}
```

Entendendo o código:

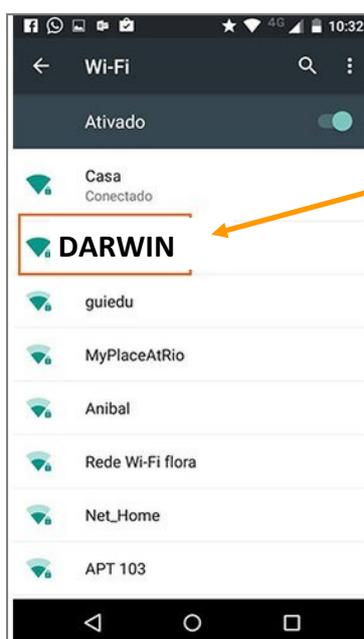
A função **AtivarWifi()** realiza todas as operações necessárias para que a placa Darwin acesse a sua rede local.

Antes de executar o programa ao lado, certifique-se de que o **display esteja funcionando**, pois ele será necessário para apresentar algumas informações importantes de configuração.

No capítulo anterior você já aprendeu como isso pode ser feito.

Passo 3: Ativando a CPU Darwin como ponto de acesso (Access Point)

Após enviar o programa acima para a placa Darwin, nossa placa se transformará em um **Ponto de Acesso** (Access Point). Utilizando seu celular, conecte-se na rede criada pela placa Darwin. Caso seja solicitada a senha, digite **12345678**.



Após estar conectado na rede do **DARWIN**, abra o navegador WEB do seu celular, e digite o endereço IP:

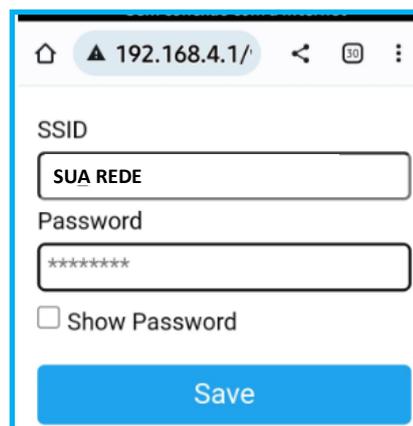
192.168.4.1

Você terá uma tela como a mostrada ao lado.



Agora que já estamos dentro da placa Darwin, podemos informar os dados de conexão com a sua **rede local**.

- Clique no botão **Configurar Wifi**
- Escolha a **rede desejada**
- Digite a **senha** da sua rede.
- Clique no botão **Save**



Pronto!

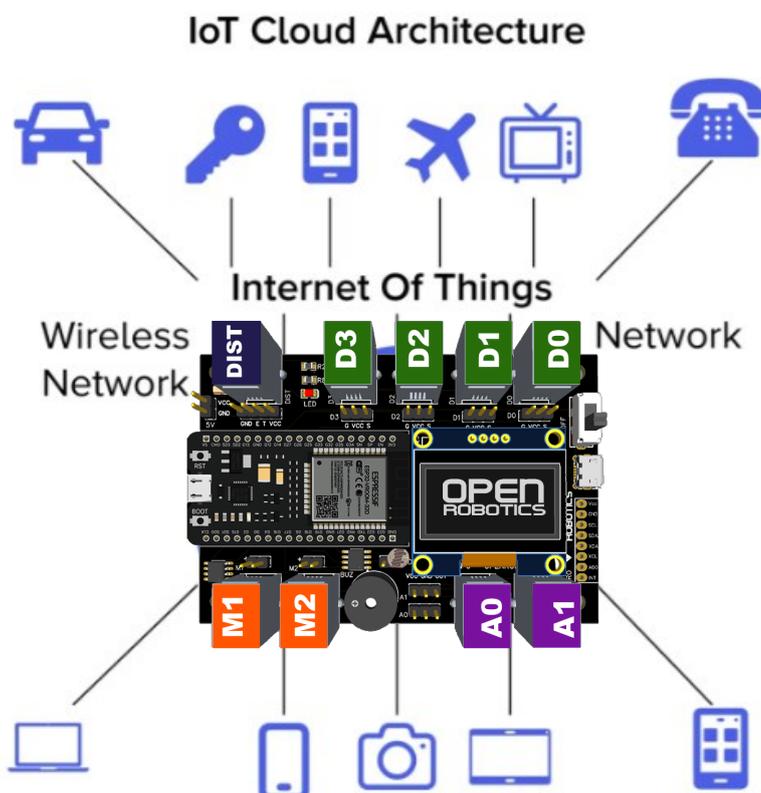
Da próxima vez que você desligar e religar a sua placa Darwin, **ele já vai acessar a sua rede local**.

E da próxima vez que quiser conectar nessa rede, o que vai acontecer?

Não se preocupe, sempre que esta rede estiver disponível, ele já vai conseguir e conectar, uma vez que ele memorizou as informações de login nesta rede.

Mas, caso você precise se conectar a outra rede, sem problema, basta repetir o procedimento para deixar várias redes configuradas ao mesmo tempo.

Veja que a partir deste momento a placa Darwin está pronta **para se conectar aos mais modernos aparelhos I.o.T** espalhados pelo mundo. Veremos isso nos próximos capítulos.



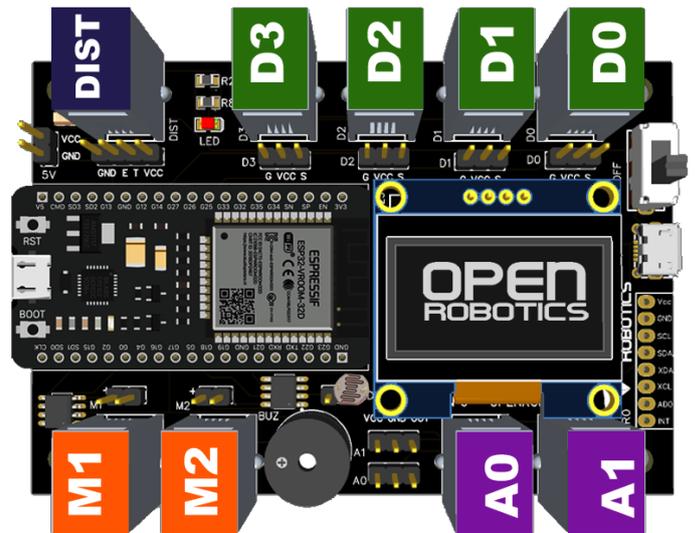
Controle por aplicativo e Bluetooth

Agora vamos ver como podemos utilizar um **aplicativo de celular** para controlar a placa Darwin.

A primeira parte do nosso trabalho será programar a placa Darwin para entrar no modo de controle por **Bluetooth**. Na segunda parte vamos instalar um App no celular para que possa enviar comandos para o robô.

Programação:

```
#include <darwin6.h>
#include <darwin6_display.h>
String msg = "";
void setup() {
  AtivarBluetooth("DARWIN");
  AtivarDisplay();
  LimparDisplay();
  EscreverDisplay(0,0,2, " BLUETOOTH");
  EscreverDisplay(0,25,2," DARWIN ");
}
void loop()
{
  valorRecebido = (char)SerialBT.read();
  if (BluetoothEstaRecebendo() == Sim)
  {
    msg = msg + valorRecebido;
    LimparDisplay();
  }
  else
  {
    EscreverDisplay(0,10,2,msg);
    msg = "";
  }
}
```



Entendendo o código:

O programa ao lado permite ativar o modo Bluetooth da placa Darwin.

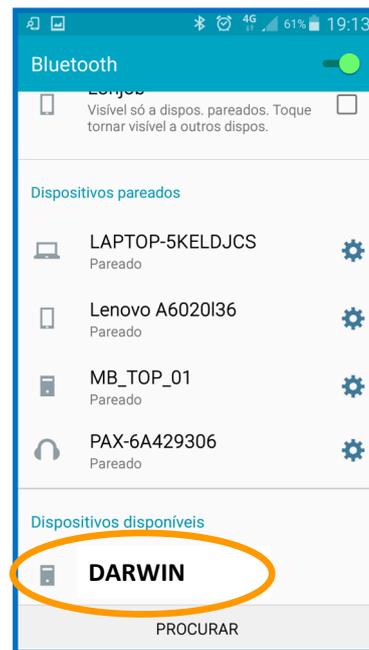
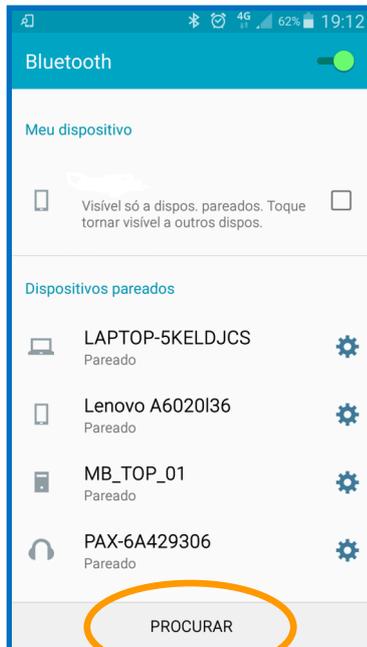
Após enviar para o robô, ele vai permitir que, a partir de um aplicativo terminal, você consiga enviar comandos que serão recebidos pelo robô.

Passo 1. Parelar o celular Android ao módulo de Bluetooth .

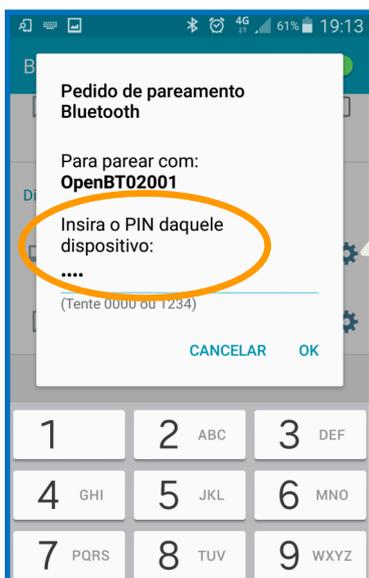
Comece ligando a placa **Darwin**

- No seu celular, **ative a função Bluetooth** e acesse a tela de configurações onde aparecem os dispositivos que já estão pareados ao seu celular.

Acompanhe a sequência de telas abaixo para finalizar o processo:

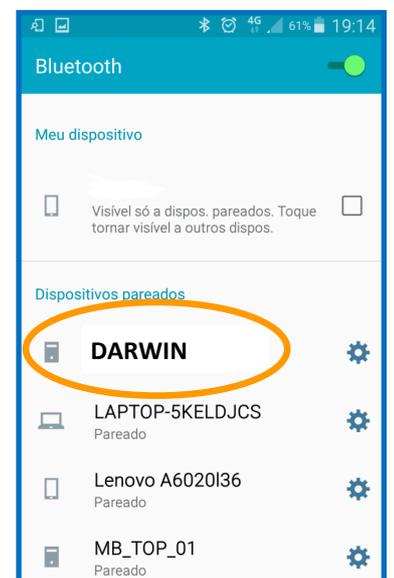


Vai aparecer um novo dispositivo na lista **Dispositivos disponíveis**. Toque no dispositivo **DARWIN** para iniciar o pareamento.



Informe a senha, que por padrão é **1234**. Depois toque em **OK**.

O dispositivo aparecerá como pareado.

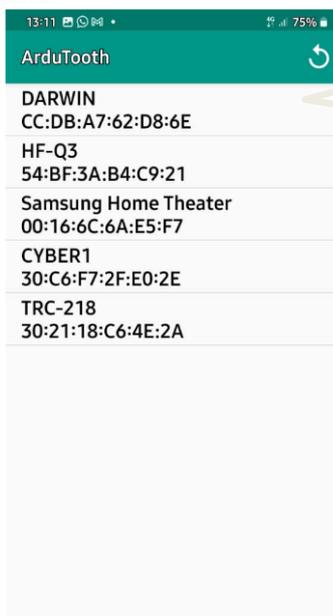


Passo 2. Instalando o Aplicativo e enviando comandos para a CPU

Agora que você já parou o seu aparelho com a placa Darwin, chegou a hora de instalar o App que irá controlar a placa. Você poderá utilizar diversos tipos de aplicativos, pode até mesmo desenvolver o seu App utilizando o [AppInventor do MIT](#), ou utilizar o aplicativo chamado ArduTooth que permite que você envie comandos para a placa Darwin.



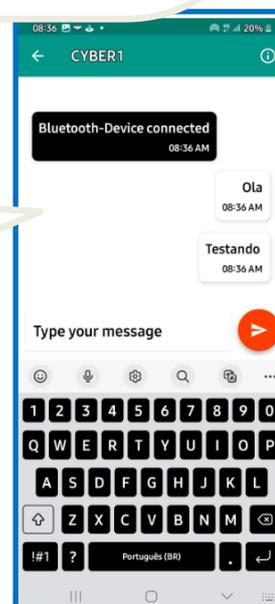
Procure na loja de aplicativos pelo App **ArduTooth** e instale ele no seu celular ou tablet.



Após instalar, entre no App, certifique-se de que o programa que ativa o Bluetooth já tenha sido enviado para na placa Darwin.

Após parear o seu aparelho com a placa Darwin, ela irá aparecer no App.

Digite um texto qualquer, e espaço no final da mensagem. Quando enviar, veja que a mensagem irá aparecer na tela OLED da placa.

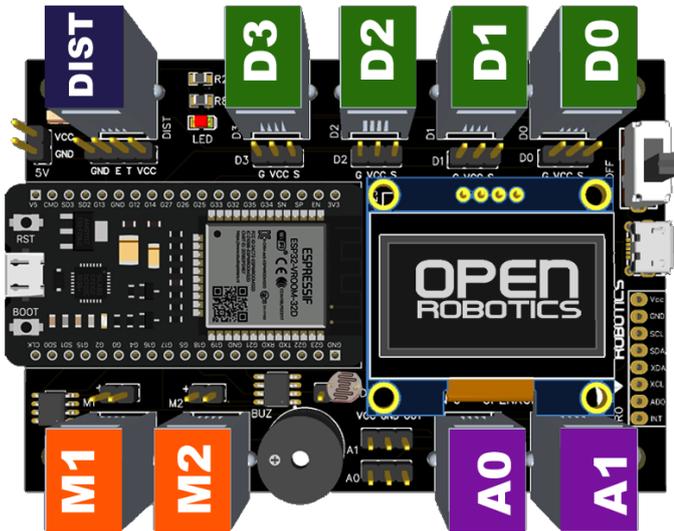


Onde utilizar?

Você poderá escrever programas que recebam comandos via Bluetooth e ativem ou desative portas digitais e motores da placa Darwin. Poderá também controlar servo motores e fazer a leitura de diversos sensores.

Conectando a placa Darwin no Telegram

Para essa lição você precisa de instalar o aplicativo **Telegram** em seu aparelho de celular. Você vai ver que esse recurso é muito útil, e pode ser utilizado para você receber mensagens vindas da placa Darwin.



Passo 1: Instalando o Telegram

Acesse a loja de aplicativos do seu celular e procure pelo Telegram. Após localizar o Telegram, clique em instalar.

Passo 2: Instalando o Bot

Para que possamos nos comunicar com a placa Darwin precisamos instalar um Bot no seu Telegram. Para isso faça o seguinte:

- Abra o Telegram, e clique na Lupa para localizar o **Botfather**
- Digite **Botfather**
- Escolha a primeira opção, como mostrada ao lado.



Agora vamos criar nosso robô para que receba nossas mensagens.

Esse robô inclusive pode ser compartilhado com outros usuários, de maneira que eles também recebam as mensagens da placa Darwin.

Passo3: Criando o Bot

Acesse a loja de aplicativos do seu celular e procure pelo Telegram. Após localizar o Telegram, clique em instalar.

- Digite **/newbot** e envie

Agora vamos dar um nome para o robô.

- Digite **Mensagens Cyber1**
(Ou um nome de sua preferência)

Por último precisamos informar um nome único para o nosso Bot, ou seja, não pode existir outro dentro do Telegram com esse nome que vamos dar.

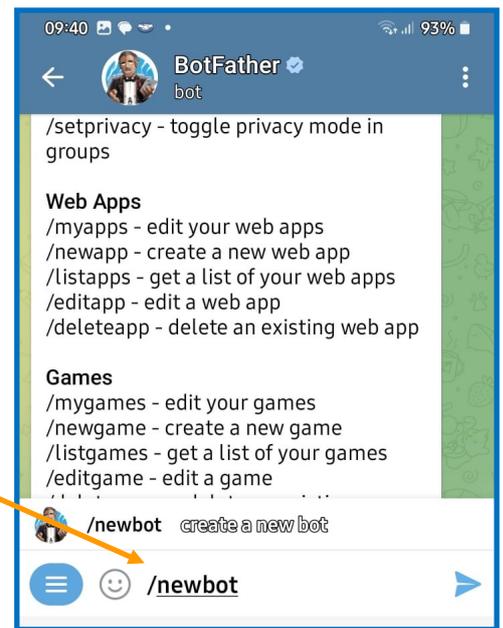
Na verdade é um código de identificação único dentro da plataforma do Telegram, por isso precisa ser exclusivo.

No exemplo colocamos **Cyber1_open_bot**.

Importante: O nome precisa necessariamente terminar com **_bot**. Não utilize o nome acima, porque certamente ele já vai existir.

Como o nome precisa ser exclusivo, experimente colocar as suas iniciais no lugar da palavra open, por exemplo:

- **Darwin_carlos_bot**
- **Darwin_meu_bot**
- **Darwin_xyzkk_bot**



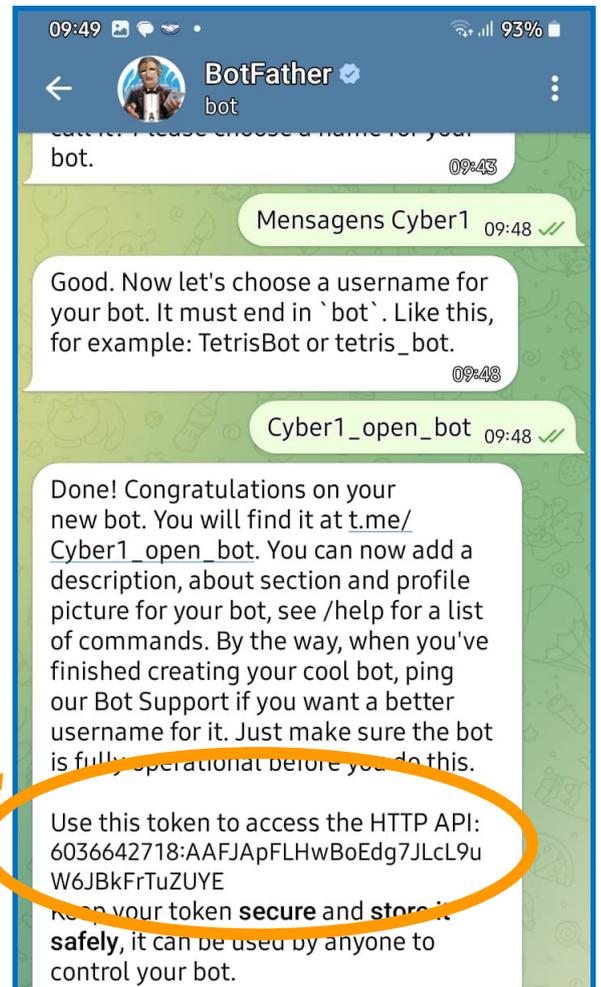
Finalizando a criação do Bot.

Após enviar o nome único para o Bot que estamos criando, você receberá uma mensagem como mostrada ao lado.

Nessa mensagem tem um código **token** muito importante.

Tome nota dessa informação, pois vamos precisar para configurar dentro da placa Darwin.

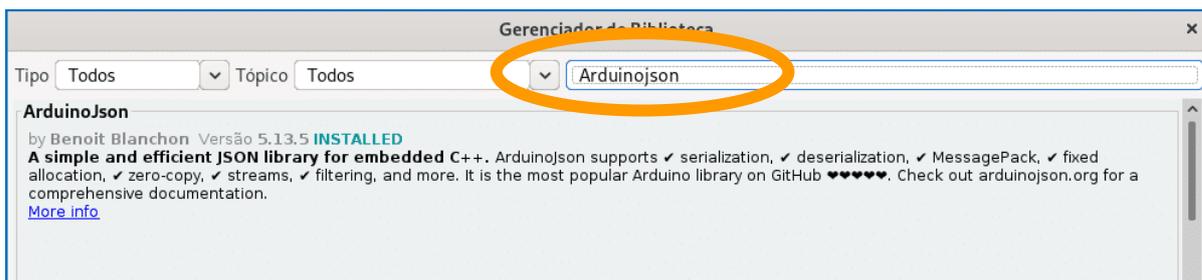
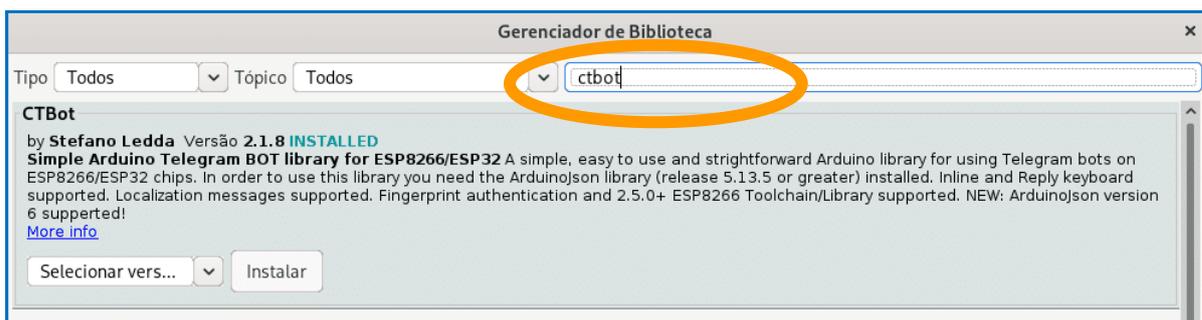
Anote aqui:



Passo 4. Instalando as bibliotecas no Arduino

Agora que nosso Instagram já está preparado, chegou a hora de programar a placa Darwin para enviar mensagens para o nosso Bot.

- No Arduino, clique em **Ferramentas, Gerenciador de Bibliotecas** e instale as duas bibliotecas mostradas abaixo, uma de cada vez:



Programando a placa Darwin

Agora que configuramos o Arduino para receber a programação, vamos aos códigos para enviar ou receber mensagens do Telegram.

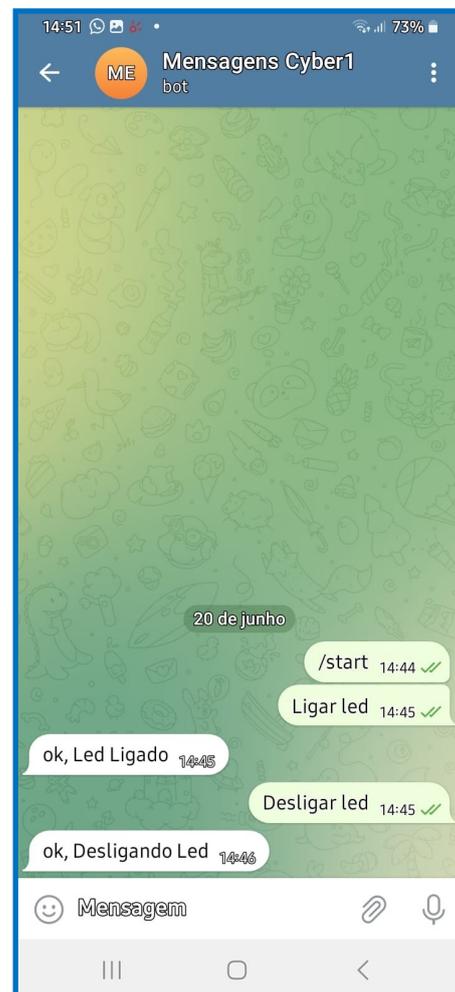
```
#include <darwin6.h>

#include <darwin6_displayh>

#include "CTBot.h"
CTBot myBot;
TBMessage msg;
String token = "DIGITE SEU TOQUEN DO INSTAGRAN AQUI";
String ssid = "DIGITE O NOME DA SUA REDE WIFI";
String pass = "DIGITE A SENHA DA SUA REDE WIFI";

void setup()
{
  AtivarDisplay();
  EsperarSegundos(5);
  LimparDisplay();
  //Inicia conexão com WiFi
  myBot.wifiConnect(ssid, pass);
  //Configura o Token do Bot
  myBot.setTelegramToken(token);
  if(myBot.testConnection())
    EscreverDisplay(0,10,2,"TELEGRAM OK!");
  else
    EscreverDisplay(0,20,1,"ERRO CONECTANDO!");
}

void loop()
{
  if(CTBotMessageText == myBot.getNewMessage(msg))
  {
    if(msg.text== "Ligar led")
    {
      myBot.sendMessage(msg.sender.id, "ok, Led Ligado");
      EscreverDisplay(0,40,2,"LED ON");
    }
    if(msg.text== "Desligar led")
    {
      myBot.sendMessage(msg.sender.id, "ok, Desligando Led");
      EscreverDisplay(0,40,2,"LED OFF");
    }
  }
}
```



Entendendo o código:

Veja que a mensagem que chega, será armazenada dentro de `msg.text`, ou seja, manipulando o valor desta variável (que contém a mensagem) você pode escrever dezenas de possibilidades em seu programa, por exemplo, ligando LED, ligando o display, ligando a sirene, mostrando no display a mensagem recebida, dentre outras infinitas possibilidades.

Resumo dos comandos e funções

LigarSirene(Parâmetro)

Permitir ativar a sirene conectada na porta digital informada no parâmetro

Parâmetro: D0, D1, D2, D3, D13

DesligarSirene(Parâmetro)

Permitir desativar a sirene conectada na porta digital informada no parâmetro

Parâmetro: D0, D1, D2, D3, D13

BotaoNaPorta(Parâmetro)

Permite ativar um botão na porta digital informada no parâmetro.

Parâmetro: D0, D1, D2, D3

LigarLed(Parâmetro)

Permitir ativar o LED conectado na porta digital informada no parâmetro

Parâmetro: D0, D1, D2, D3, D12

DesligarLed(Parâmetro)

Permitir desativar o LED conectado na porta digital informada no parâmetro

Parâmetro: D0, D1, D2, D3, D12

EsperarSegundos(Parâmetro)

Permite aguardar um tempo no processamento.

Parâmetro: Valor inteiro em segundos.

ServoNaPorta(Parâmetro)

Permite indicar que um servo motor será colocado em uma porta digital.

Parâmetro: D1, D2, D3

MoverServo(Parâmetro 1, Parâmetro 2)

Permite girar um servo motor na porta digital, movendo-o para um determinado ângulo entre 0 e 180.

Parâmetro 1: D1, D2, D3

Parâmetro 2: 0 a 180

Resumo dos comandos e funções

TocarSom(Parâmetro 1, Parâmetro 2)

Permite tocar um tom na porta onde a sirene estiver conectada.

Parâmetro 1: Indica a porta a digital onde está a sirene

Parâmetro 2: Indica o tom a ser tocado (valor entre 0 e 1024)

Silenciar(Parâmetro)

Permite silenciar a sirene na porta digital indicada.

Parâmetro: D1, D2, D3, D12

LerBotao(Parâmetro)

Permite ler o valor do botão presente na porta digital

Parâmetro: D1, D2, D3

AtivarComunicacao()

Permite ativar as portas de comunicação serial ou bluetooth

Parâmetro: —

EnviarParaComputador(Parâmetro)

Permite enviar uma informação para ser lida no computador

Parâmetro: Valor a ser enviado para o computador

EnviarParaBluetooth(Parâmetro)

Permite enviar uma informação para ser lida no celular via bluetooth

Parâmetro: Valor a ser enviado via bluetooth

LerLuz(Parâmetro)

Permite realizar a leitura da quantidade de luz no sensor de Luz, retornando claro ou escuro.

Parâmetro: A0, A1, A2

Retorno: 0 (Ausência de luz) 1 (Presença de luz)

Resumo dos comandos e funções

LerLuzInterna(Parâmetro)

Permite realizar a leitura da quantidade de luz no sensor de Luz, retornando claro ou escuro.

Parâmetro: —

Retorno: 0 (Ausência de luz) 1 (Presença de luz)

LerInclinacao(Parâmetro)

Permite ler a inclinação presente no sensor de inclinação (Horizontal ou vertical)

Parâmetro: A0, A1, A2

Retorno: 1 (para horizontal) 0 (para vertical)

LerSom(Parâmetro)

Permite ler a quantidade de som presente no sensor de som.

Parâmetro: A0, A1, A2

Retorno: 1 (com algum ruído) 0 (silêncio)

LerMagnetico(Parâmetro)

Permite ler o sensor magnético

Parâmetro: A0, A1, A2

Retorno: 1 (Com sinal magnético) 0 (Sem sinal magnético)

ReceberDoComputador()

Permite receber algum valor enviado pelo computador.

Parâmetro: —

ReceberDoBluetooth()

Permite receber algum valor enviado pelo computador.

Parâmetro: —

SortearNumero(Parâmetro 1, Parâmetro 2)

Retornar um valor sorteado entre os valores do parâmetro 1 e do Parâmetro 2

Parâmetro1: Valor inteiro

Parâmetro 2: Valor Inteiro maior que o parâmetro 1.

Resumo dos comandos e funções

AtivarMotores()

Permitir ativar os motores que estão nas portas M1 e M2.

Parâmetro: Não necessário

MotorM1(Parâmetro)

Permite ligar o motor na porta M1, indicando a direção a ser seguida pelo mesmo.

Parâmetro: Frente, Re, Parar

MotorM2(Parâmetro)

Liga o motor na porta M2, indicando a direção a ser seguida pelo mesmo.

Parâmetro: Frente, Re, Parar

LeituraDaPorta(Parâmetro)

Utilizado para ler a porta analógica

Parâmetro: A0,A1,A2,A3

LerDistancia()

Retorna 1 ou 0 conforme a leitura da porta do sensor Infravermelho (DIST)

Parâmetro: Não necessário.

Retorno: Valor em centímetros

LedNaPorta(Parâmetro)

Utilizado para ativar um LED na porta digital indicada.

Parâmetro: D0, D1, D2, D3, D12

SireneNaPorta(Parâmetro)

Utilizado para ativar uma SIRENE na porta digital indicada.

Parâmetro: D0, D1, D2, D3, D12