


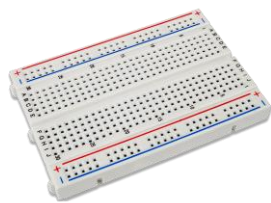






Pretende-se que **realizes na prática** a atividade do **sensor ultrassónico** realizada em prática simulada no **Tinkercad** no guião anterior. Para isso irás necessitar do seguinte material:

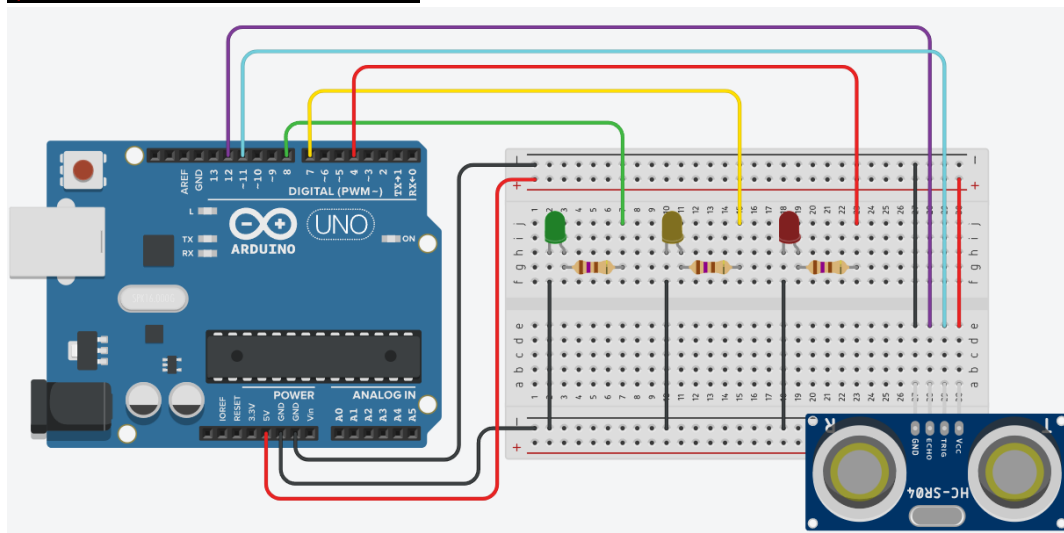
<p>Arduino UNO</p> 	<p>Resistências</p> 
<p>LEDs – vermelho, verde e amarelo</p> 	<p>Breadboard</p> 
<p>Cabos tipo Dupont</p> 	<p>Sensor Ultrassónico HC-SR04</p> 



No teu **diário de bordo**, cria um novo **título** no **final** do **documento** de nome **Arduino Laboratório 5 – Sensor Ultrassónico**.



Começa por **montar** os seguintes **componentes** na **Breadboard**, reproduzindo fisicamente o que **fizeste** em **Tinkercad** no **guião** passado: **Atenção: repara que por questão de comodidade viramos o sensor ultrassónico ao contrário no Tinkercad, pelo que terás de verificar no teu sensor físico as ligações de acordo com a orientação que deres ao sensor quando o encaixas na Breadboard.**

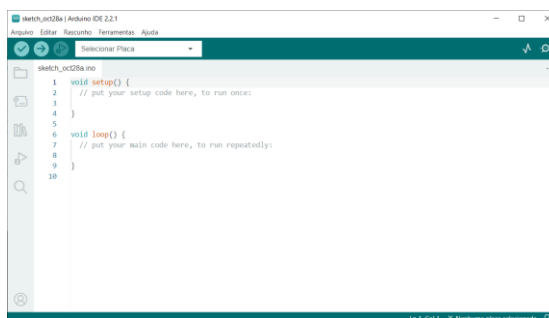




Utiliza as resistências que te parecerem mais adequadas de acordo com as que te forem disponibilizadas pelo teu professor.



Abre o IDE do Arduino.



Neste programa terá de ser possível realizar o mesmo da simulação do Tinkercad. Relembrando: conforme a distância os LEDs devem ligar-se ou desligar-se. O vermelho liga-se caso a distância for inferior a 20cm, o amarelo liga-se caso a distância for maior 20cm e menor que 100cm e o verde se a distância for maior que 100cm.



Começa por definir os pinos do Arduino com que vais trabalhar.

Para os LEDs necessitamos de 3 pinos:

Pino 4 para o vermelho;

Pino 7 para o amarelo;

Pino 8 para o verde.

Para o sensor ultrassónico necessitamos de 2 pinos:

Pino 11 para o Trigger;

Pino 12 para o Echo.

Sabendo que os pinos são sempre definidos antes da função Setup() e que a sua sintaxe é a seguinte:

```
#define Pino N°  
Ou  
const int Pino=N°;
```

Exemplo:

```
#define triggerPin 11
```

Ou

```
const int redPin=4;
```

Define os restantes pinos a utilizar.



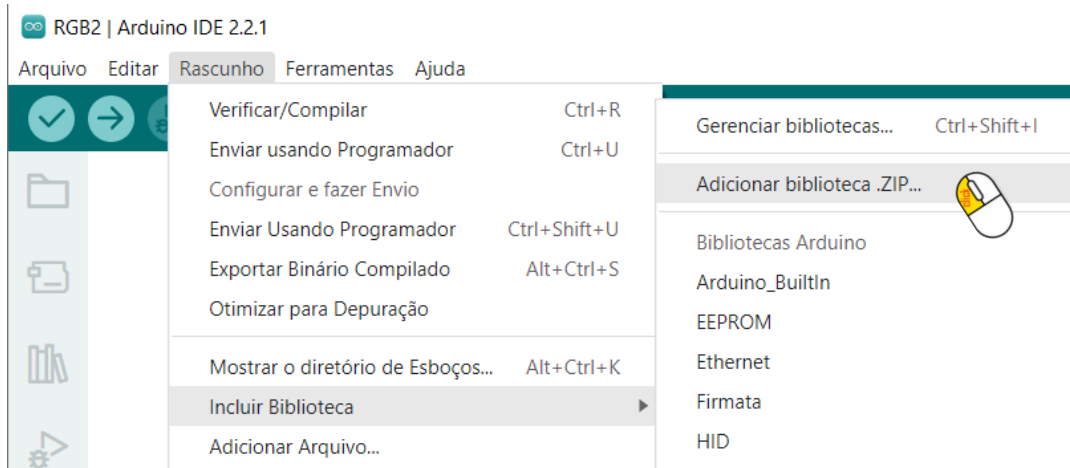
Depois de definirmos os pinos temos de indicar se vão ser de saída ou entrada. Como queremos acender Leds queremos escrever no pino logo são de saída (OUTPUT). Para isso na função **setup()**, a que só é lida uma vez ao iniciar o programa dizemos o que cada um dos pinos definidos será. Deixo aqui um exemplo para o pino do LED vermelho:

```
pinMode(redPin, OUTPUT); //função que define se a porta digital é de output ou input
```

Completa com os outros pinos. Os pinos do sensor ultrassónico não necessitam de ser definidos como entrada ou saída, apenas os dos LEDs.



Agora vamos tratar do **sensor ultrassónico**. Para o **Arduino** poder **trabalhar** com o **sensor ultrassónico** é necessário **adicionar** uma **biblioteca** que **permitirá** dispor de **funções** específicas deste componente e assim **programa-lo**. Começa por **descarregar** a **biblioteca aqui** e de **seguida** a **adiciona-a** ao IDE do **Arduino**:



Como **vimos** na aula **teórica**, este sensor pode ser **utilizado** com **funções específicas**, sem a **necessidade** de **efetuar cálculos**. O processo para utilizar este **sensor** é sempre o **mesmo**. Começa por **adicionar** a seguinte **linha** de **código** como **1ª linha**:

```
1 #include <Ultrasonic.h>
```

Esta linha **permite** ter **acesso** à biblioteca **Ultrasonic.h** que **instalaste anteriormente**, ficando à nossa disposição **enumeras funcionalidades**.



Antes da função **Setup()**, **depois** da zona de código onde **definiste** os **pinos**, **acrescenta** a seguinte linha de **código**:

```
Ultrasonic ultrassonico(triggerPin,echoPin);
```

Com esta linha é **criado** um **objeto** de nome **ultrassonico** do tipo **Ultrasonic**. Para quem tem poucas bases de programação por objetos, parece um código estranho... eu sei! Porém, tal como referi o processo é sempre o mesmo e desta forma **podes** **aceder** às tais funcionalidades. Como? Uma vez com o objeto criado basta **utilizá-lo** da **seguinte forma**:

```
Nome_objeto.função
```



Pronto para **utilizar**? Como temos de saber o **tempo** que **demora** o **sinal** a **embater** no **obstáculo** e **voltar** e também **converter** esse **valor** em **cm**, mas **sem** **efetuar cálculos** (para ser mais simples) vamos **utilizar duas funções**:

```
ultrassonico.timing() e ultrassonico.convert()
```



Para **guardar** estes **valores** é melhor utilizar **variáveis**. **Define** duas variáveis:

```
float distancia;  
long tempo;
```



Estes **valores** têm de estar **constantemente** a ser **verificados** para **saber** se **algum obstáculo** se **aproxima**. Então serão **atualizados** na **função** do **programa principal**, ou seja na função **loop()**. **Acrescenta** dentro de **loop()** as seguintes **linhas** de **código**:

```
tempo = ultrassonico.timing();  
distancia= ultrassonico.convert(tempo,Ultrasonic::CM);
```



Para já **verifica** se tens o **código semelhante ao meu**:

```
#include <Ultrasonic.h>
const int redPin=4, yellowPin=7, greenPin=8;
#define triggerPin 11
#define echoPin 12
Ultrasonic ultrassonico(triggerPin,echoPin);
float distancia;
long tempo;

void setup() {
  pinMode(redPin,OUTPUT);
  pinMode(yellowPin,OUTPUT);
  pinMode(greenPin,OUTPUT);
}

void loop() {
  tempo = ultrassonico.timing();
  distancia= ultrassonico.convert(tempo,Ultrasonic::CM);
```

Como podes **verificar** o **sensor** dá-te a **distância** a cada **loop()**. O que tens de **fazer** é **testar** a **distância** com uma **estrutura** de **decisão** e **conforme** o **valor** dessa **distância** **ligas** ou **desligas** os **LEDs**. Deixo-te a **sintaxe** da **estrutura** de **decisão** **if**:

```
if (condição1){
  se condição1 verdadeira faz isto e sai fora do if;
} else if (condição2) {
  se condição2 verdadeira faz isto e sai fora do if;
} else if (condiçãoN) {
  se condiçãoN verdadeira faz isto e sai fora do if;
} else {
  faz isto (apenas) caso nenhuma das condições
  anteriores se verifique;
}
```

Para teres **duas condições** dentro do **IF** podes **fazer**:

```
if (condição1 && condição2){(...)}
```

Deixo-te o **código** para o **LED vermelho**:

```
if(distancia<=20.0){
  digitalWrite(redPin,HIGH);
  digitalWrite(yellowPin,LOW);
  digitalWrite(greenPin,LOW);
```

Completa com o **código** que te **permita acender** o **LED amarelo** e **verde**.



Adiciona ao final do **loop()** um **delay()** para que o **loop** se realize a **cada 100ms**:

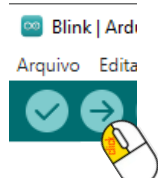
```
delay(100);
```



Adiciona ainda **condições iniciais** para que os **LEDs** estejam todos **desligados** no **início** do **programa**. Onde deves **inserir** este **código**?



Caso não tenha erros clica agora no botão para enviar o programa para o Arduino.





Faz um **pequeno vídeo** com o **telemóvel do circuito + Arduino** com os **LEDs a ligar/desligar** conforme a **distância** de um **objeto** ao **sensor**. Envia para o teu **Drive** para a pasta **Arduino - Vídeos**. **Altera o nome** do ficheiro de vídeo para **5 – Ultrassónico**.



Tira também uma foto e cola no teu **diário de bordo**.



Guarda o programa, no IDE do Arduino com o nome **5Ultrassónico**. Envia este **ficheiro** para o teu **drive** para a pasta **Arduino - Programas**



Chama o teu professor para avaliar.