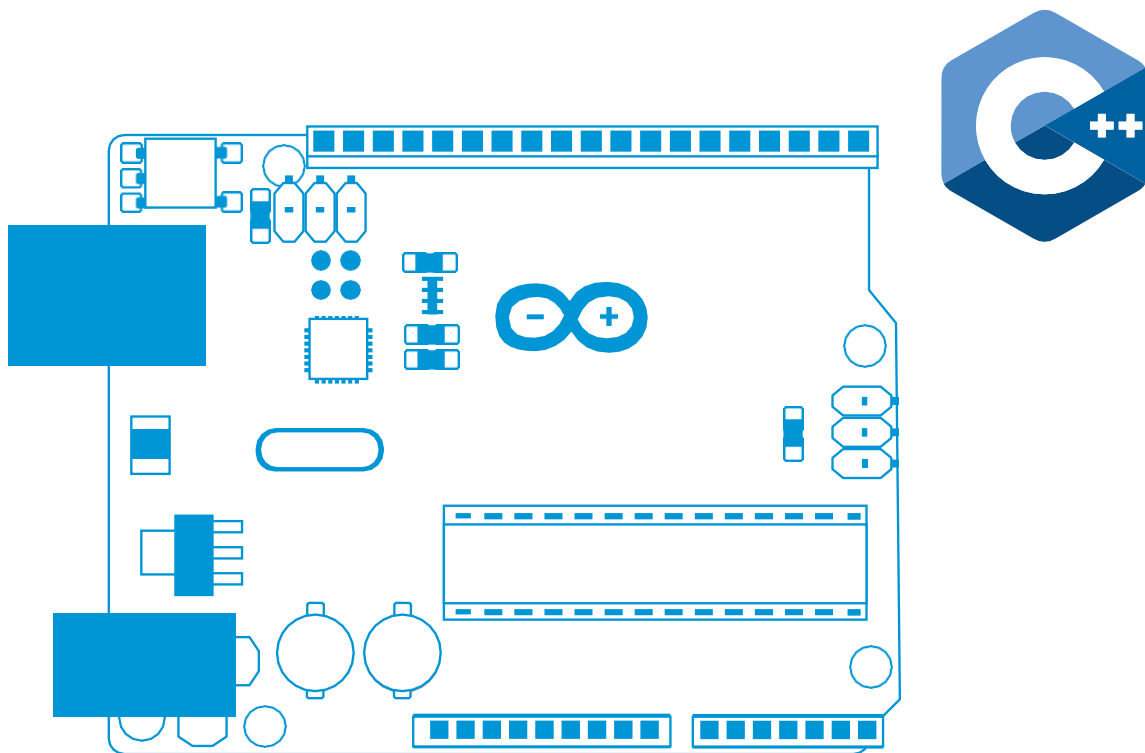
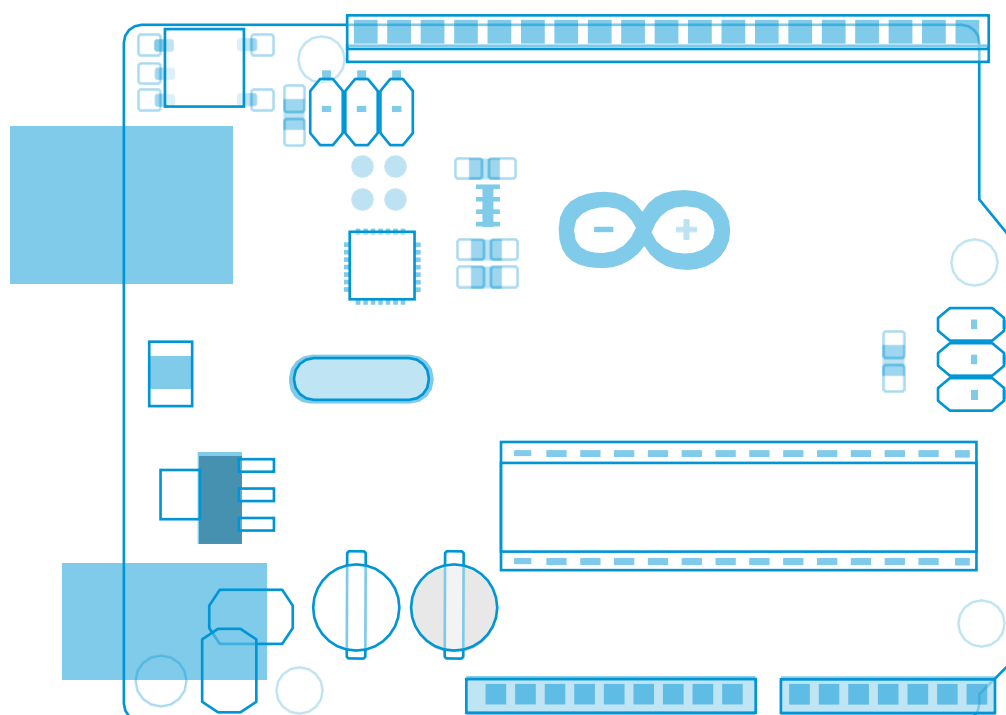
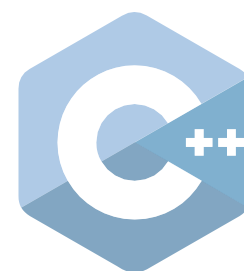


# Ensinar com o espaço

## → CONHECE O ARDUINO!

Introdução à computação com o Arduino utilizando C++





Notas	página 3
Sumário das atividades	página 4
Atividade 0: Vamos começar	página 5
Atividade 1: Faz-me piscar!	página 8
Atividade 2: Pisca um S.O.S.!	página 11
Atividade 3: Medindo a temperatura	página 12
Atividade 4: Medindo a pressão	página 16
Atividade 5: Medindo a altitude	página 18
Fichas de trabalho	página 19
Links	página 34

Ensinar com o espaço – conhece o arduino | T04.1  
[www.esa.int/education](http://www.esa.int/education)

Departamento de Educação da ESA agradece feedback  
e comentários [teachers@esa.int](mailto:teachers@esa.int)  
**Produzido pela ESA educacional**

Copyright 2018 © Agência Espacial Europeia

## → CONHECE O ARDUINO!

### Introdução à computação com o Arduino utilizando C++

#### Notas

**Nível etário:** 14-20 anos

**Área curricular:** Programação, Eletrónica

**Complexidade:** Média

**Duração:** 90-120 minutos

**Localização:** dentro de casa

**Inclui a utilização de:** software e hardware  
Arduino

**Palavras chave:** Arduino, Sensor, Código

#### Breve descrição

Os alunos vão explorar tecnologia usada no espaço através da ferramenta Arduino. Vão construir circuitos para fazer piscar um LED e para medir a temperatura, a pressão e a altitude. A programação básica em C++ será introduzida utilizando software do Arduino IDE (Ambiente de Desenvolvimento Integrado). Este conjunto de atividades pode ser o ponto de partida para uma futura participação na [Competição CanSat](#).

#### Objetivos de aprendizagem

- Melhorar competências analíticas
- Desenvolver a compreensão de uma linguagem de programação
- Desenvolver o conhecimento sobre montagem de circuitos
- Compreender como podem ser utilizados sensores para obter informação
- Melhorar competências do trabalho em equipa

**Atividades – Descrição e resultados esperados**

	<b>Título</b>	<b>Descrição</b>	<b>Resultado esperado</b>	<b>Pré-requisitos</b>	<b>Duração</b>
1	Vamos começar	Introdução aos componentes utilizados ao longo da atividade.	Os alunos vão-se familiarizar com os componentes necessários e com as suas funções.	Conhecimento básico de componentes eletrónicos.	10 minutos
1	Faz-me piscar!	Os alunos vão construir o seu primeiro circuito Arduino.	Os alunos vão conseguir controlar um LED com o código que escreveram.	Conclusão da atividade anterior.	15 minutos
2	Pisca um S.O.S	Os alunos vão aprender a enviar mensagens mais complexas usando um LED.	Os alunos vão conseguir escrever o código e programar o LED para enviar um S.O.S.	Conclusão das atividades anteriores.	15 minutos
3	Medindo a temperatura	Os alunos vão utilizar um sensor de temperatura para medir a temperatura do ambiente circundante.	Os alunos vão ser capazes de construir circuitos mais complexos e utilizá-los para fazer medições relativas ao seu meio ambiente.	Conclusão das atividades anteriores.	20 minutos
4	Medindo a pressão	Os alunos são ensinados a utilizar um sensor de pressão para medir a pressão atmosférica do local.	Os alunos vão ser capazes de construir circuitos mais complexos e utilizá-los para fazer medições relativas ao seu meio ambiente.	Conclusão das atividades anteriores.	20 minutos
5	Medindo a altitude	Os alunos são ensinados a determinar a altitude a partir de medições da pressão atmosférica local.	Os alunos vão conseguir, com confiança, construir circuitos e manipular códigos para fazer medições relativas ao seu meio ambiente.	Conclusão das atividades anteriores.	20 minutos

## → Atividade 0: Vamos começar

### Introdução

O Arduino é uma plataforma aberta utilizada para construir uma ampla variedade de projetos eletrônicos, incluindo a possibilidade de simular missões espaciais reais como a ExoMars (<http://exploration.esa.int/mars/>).

O Arduino integra uma placa de circuito programável, (hardware) e software ou IDE (Ambiente de Desenvolvimento Integrado) que corre num computador.

O hardware e o software do Arduino foram projetados por artistas, designers, amadores, hackers e pessoas interessadas em criar objetos interativos e ambientes. O Arduino pode ser utilizado para interagir com botões, LEDs, sensores, motores, alto-falantes, unidades de GPS, câmaras, internet, e até o seu smart-phone ou TV! Esta flexibilidade, juntamente com o facto do software do Arduino ser gratuito, e de as placas de hardware e os sensores serem relativamente baratos, bem como serem ambos, o software e o hardware, intuitivos, conduziu a uma grande comunidade de utilizadores ([www.arduino.cc](http://www.arduino.cc)).

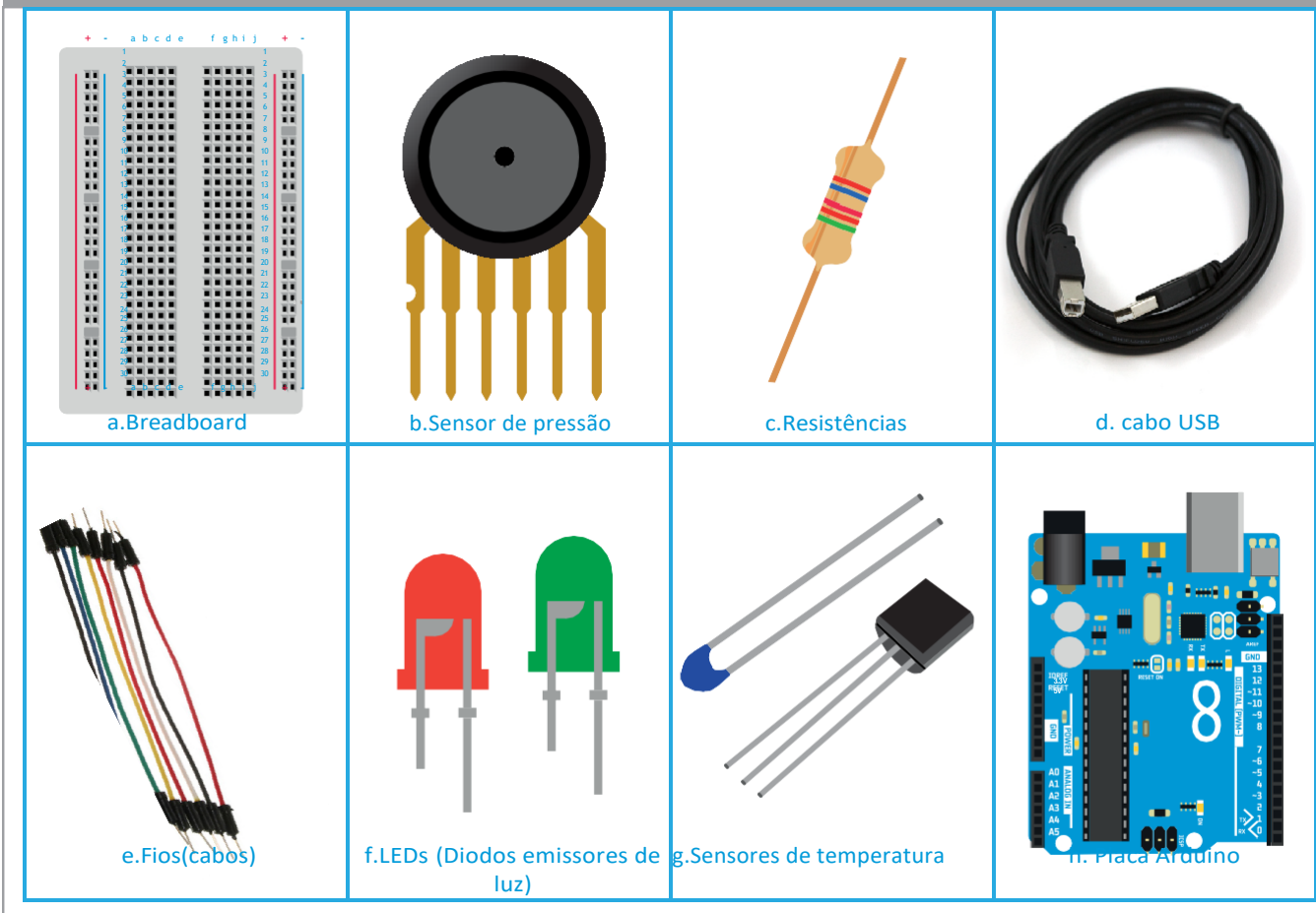
### Hardware

Para esta atividade vamos utilizar o Arduino Uno (Figura A1, caixa h). O Arduino pode fornecer voltagem aos seus pinos (terminais metálicos) e ler a respetiva voltagem. Observe o número impresso junto a cada pino, com a **GND** (terra) a atuar como polo negativo e **Vin** ou **5V** a atuarem como polos positivos (como os polos positivo e negativo de uma bateria).

Vamos utilizar os seguintes componentes que se veem na Figura A1:

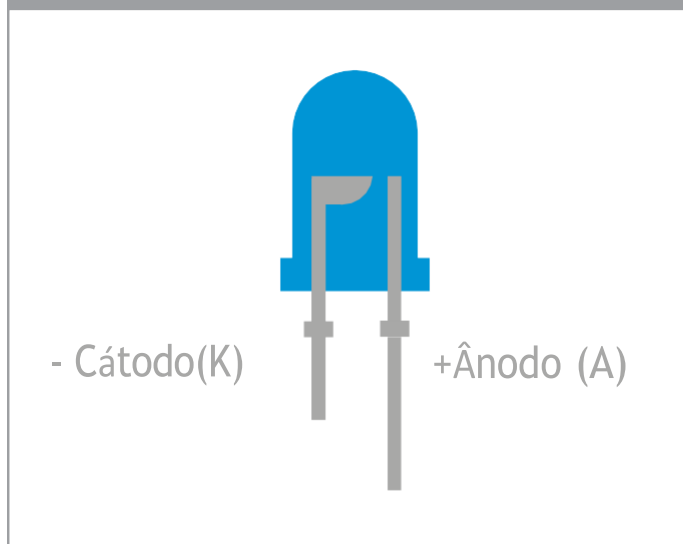
- a. Breadboard:** estrutura de suporte para ligar facilmente componentes eletrônicos.
- b. Sensor de pressão MPX4115A:** Componente que mede a pressão absoluta do ambiente.
- c. Resistências de 220Ω e 10kΩ:** São para controlar o fluxo de corrente. A resistência mede-se em ohms ( $\Omega$ ). As resistências têm faixas coloridas que indicam o seu valor.
- d. Cabo USB:** Usado para ligar o Arduino ao computador e/ou a uma fonte de energia.
- e. Cabos:** Usados para ligar eletricamente os componentes.
- f. LEDs (Diodo emissor de luz):** Componentes eletrônicos que geram luz quando a corrente elétrica os atravessa.
- g. Sensores de temperatura:** Um sensor analógico (tal como o LM35) e um termistor (é uma resistência que é particularmente sensível à temperatura).
- h. Placa do Arduino:** Funciona como um simples computador que contém um processador, memória e alguns pinos de entrada /saída.

Figura A1



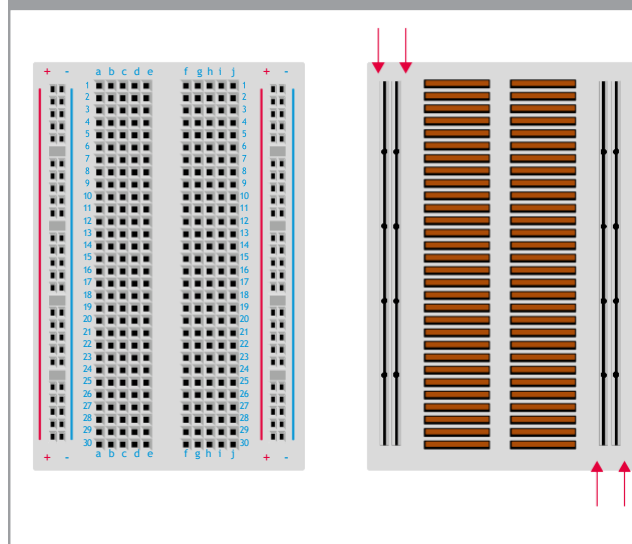
↑ Componentes do kit básico do Arduino

Figura A2



↑ LED: Observe que o LED tem um terminal comprido e outro mais curto. O comprido chama-se ânodo e o curto cátodo.

Figura A3



↑ Breadboard: Observe que todos os pinos de cada fila estão ligados.

Quando ligar o Arduino, o seu computador deverá reconhecê-lo e iniciar o processo de instalação. Se tiver problemas a instalar as drives pode encontrar instruções adicionais aqui:

[www.arduino.cc/en/Guide/ArduinoUno](http://www.arduino.cc/en/Guide/ArduinoUno)

\*Por favor assegure-se que seleciona a placa correta (tente Genuino Uno) e a porta correta (COM) no menu das ferramentas!

## Software

O software do Arduino pode ser facilmente descarregado e instalado aqui: [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software)

Os programas que são escritos utilizando o Software (IDE) do Arduino são chamados sketches. Estes sketches são escritos no editor de texto e são guardados com a extensão .ino. O editor tem ferramentas para cortar/colar e para procurar/substituir texto. A área de mensagens, no fim da página, guarda, exporta e também apresenta os erros. A placa selecionada e a porta em série são apresentadas no canto inferior direito da janela. Os ícones da barra de ferramentas permitem-lhe verificar e carregar programas, criar, abrir, guardar os sketches e abrir o monitor em série (para visualizar a informação coletada pelos sensores). A figura A4 é uma captura de ecrã da interface gráfica do Arduino IDE.

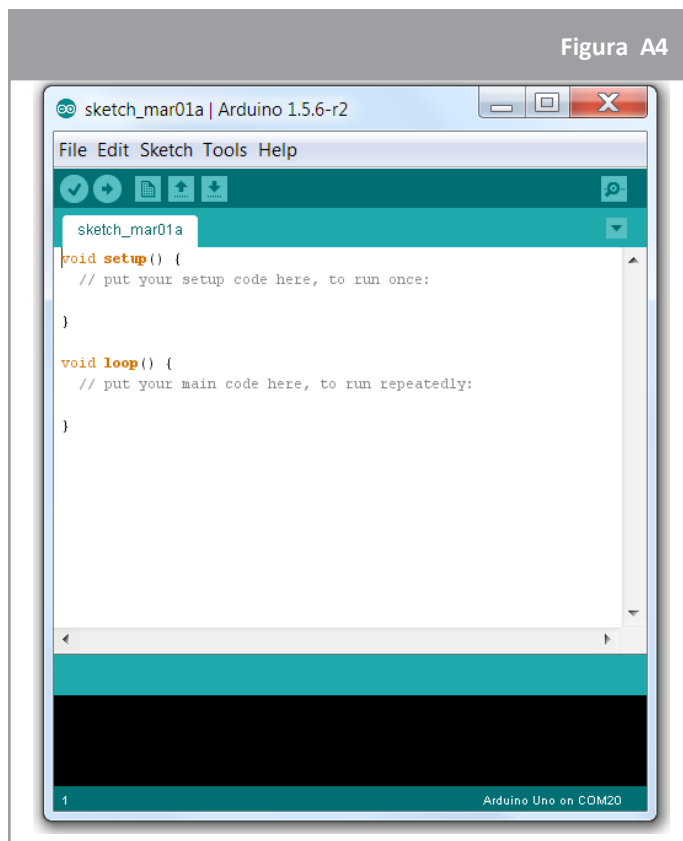








Figura A4

↑ interface do Arduino IDE

### Commandos do Arduino - Tabela A1

	<b>Verificar:</b> verifica se o seu código tem erros enquanto o compila
	<b>Carregar:</b> Compila o seu código e descarrega-o para a placa configurada.
	<b>Novo:</b> Cria um novo sketch.
	<b>Abrir:</b> Mostra um menu de todos os sketches no seu arquivo de sketches.
	<b>Guardar:</b> Guarda o seu sketch.
	<b>Monitor em série:</b> Abre o monitor, permitindo visualizar a informação obtida pelos sensores.

Comandos adicionais encontram-se nos cinco menus: File, Edit, Sketch, Tools e Help.

## → Atividade 1: Faz-me piscar!

Nesta atividade os alunos vão controlar um LED usando o Arduino. O objetivo é controlar o piscar do LED e compreender como se escreve e utiliza o código para controlar cada um dos LED. Pode ser encontrada mais informação fazendo uma pesquisa no Google: Piscar um LED com o Arduino.

### Material necessário

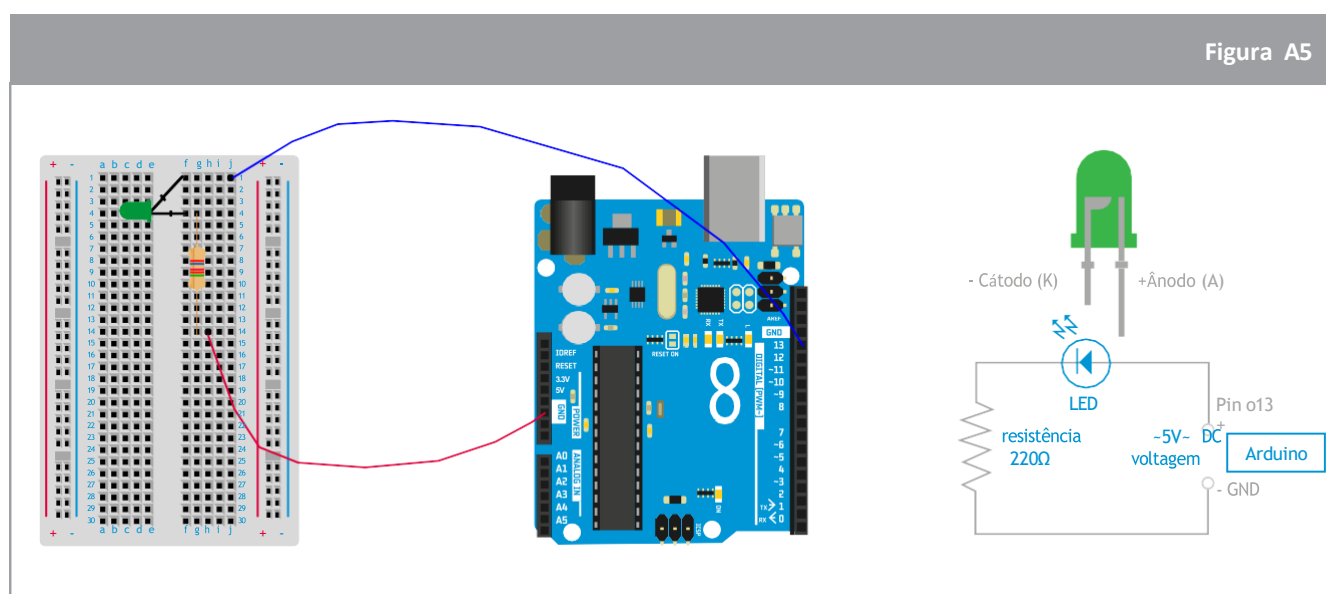
- 1 Arduino Uno
- 1 breadboard
- 1 LED (verde)
- 1 resistência de  $220\Omega$  (vermelho/vermelho/preto/preto/castanho)
- 2 fios

### Exercício

### Montagem

O circuito é montado da seguinte maneira:

- Ligue o LED verde à breadboard como mostra a Figura A5.
- Ligue a resistência de  $220\Omega$  (vermelho/vermelho/preto/preto/castanho) ao terminal mais curto do LED como mostra a Figura 5.
- Usando um fio, ligue o terminal mais comprido do LED ao pino número **13** do Arduino Uno.
- Usando um fio, ligue a resistência ao pino **GND** (Terra) do Arduino Uno.
- Por fim, ligue o Arduino UNO ao computador utilizando o cabo USB.



↑ Circuito Arduino preparado para piscar o LED verde



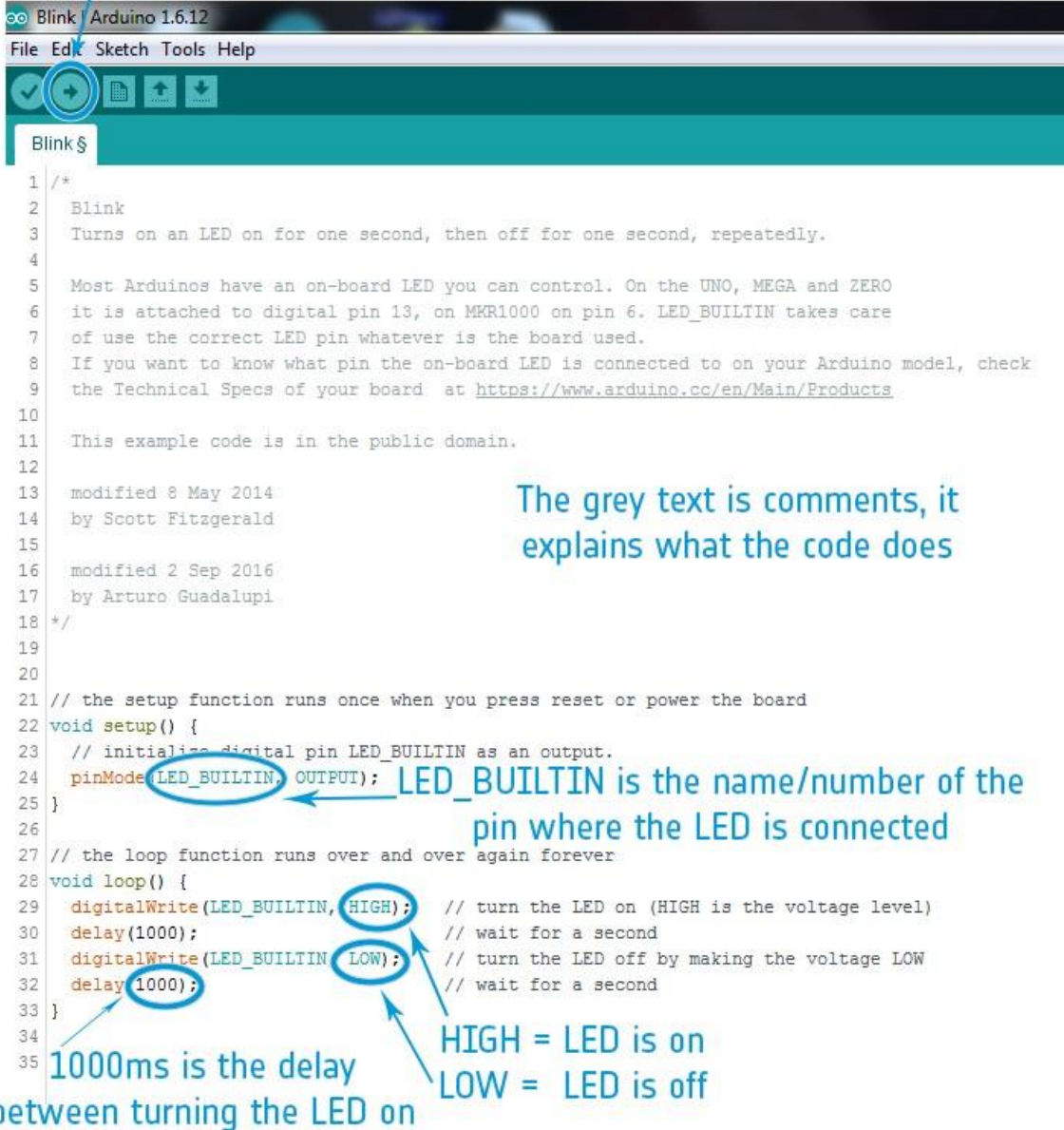
## Código

Os alunos vão agora carregar um programa para controlar o LED. Eles vão utilizar um código de exemplo que pode ser encontrado abrindo o software do Arduino (IDE) e clicando em:

File → Examples → Basics → Blink

O código para piscar aparece no ecrã como se mostra na imagem em baixo. Os comentários são exibidos a cinzento depois da barra e explicam cada linha do código.

### Upload the program to the Arduino



```
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model, check
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products.
10
11  This example code is in the public domain.
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
17  by Arturo Guadalupi
18 */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
35
```

The grey text is comments, it explains what the code does

LED\_BUILTIN is the name/number of the pin where the LED is connected

HIGH = LED is on  
LOW = LED is off

1000ms is the delay between turning the LED on

Os elementos principais deste código estão explicados em baixo:

```
void setup()
```

este comando encontra-se sempre no início quando se começa um sketch; ele inicializa as variáveis que vão ser usadas durante todo o sketch. Só vai correr uma vez, depois de se ligar ou de se fazer o reset da placa do Arduino.

```
pinMode(LED_BUILTIN, OUTPUT) -> pinMode(13, OUTPUT)
```

Configura o pino especificado para funcionar quer como entrada (input) quer como saída (output). No exemplo de piscar, substitua o **LED\_BUILTIN** pelo número do pino onde o LED está ligado, i.e. **13**.

```
void loop()
```

Este comando faz exatamente o que o seu nome sugere e vai fazer ciclos consecutivos de todos os comandos que contem, permitindo ao seu programa fazer múltiplas coisas sucessivamente.

```
digitalWrite(13, HIGH)
```

Liga o LED. O HIGH define a voltagem para aproximadamente 5V no pino 13.

```
delay(1000)
```

Coloca o programa em pausa pelo tempo indicado (em milissegundos).

```
digitalWrite(13, LOW)
```

Desliga o LED pondo a voltagem em LOW. Low significa fornecer 0V volts ao pino 13.

Uma lista completa de todos os comandos existentes pode ser encontrada numa pesquisa rápida no Google: Comandos do Arduino.

## → Atividade 2: Pisca um S.O.S.!

Nesta atividade os alunos usam o circuito Arduino construído na atividade 1. Vão controlar o LED verde para enviar uma mensagem em Morse, demonstrando assim como podiam comunicar com um rover em Marte. Para enviar um O, os alunos devem alterar o delay no código para aumentar o tempo durante o qual o LED verde está aceso. Neste caso alteramos 1000 ms para 5000 ms, mas pode ser alterado para qualquer número superior a 1000 ms. O código está na Figura A6:

Figura A6

```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
}
```

↑ Código para enviar um O em Morse

## → Atividade 3: Medindo a temperatura

Nesta atividade os alunos medem a temperatura utilizando o Arduino. O objetivo é ler a temperatura do sensor de temperatura (termistor\*) e perceber como se escreve e utiliza o código para controlar o sensor. Pode encontrar mais informação fazendo uma pesquisa rápida no Google: sensor termistor do Arduino.

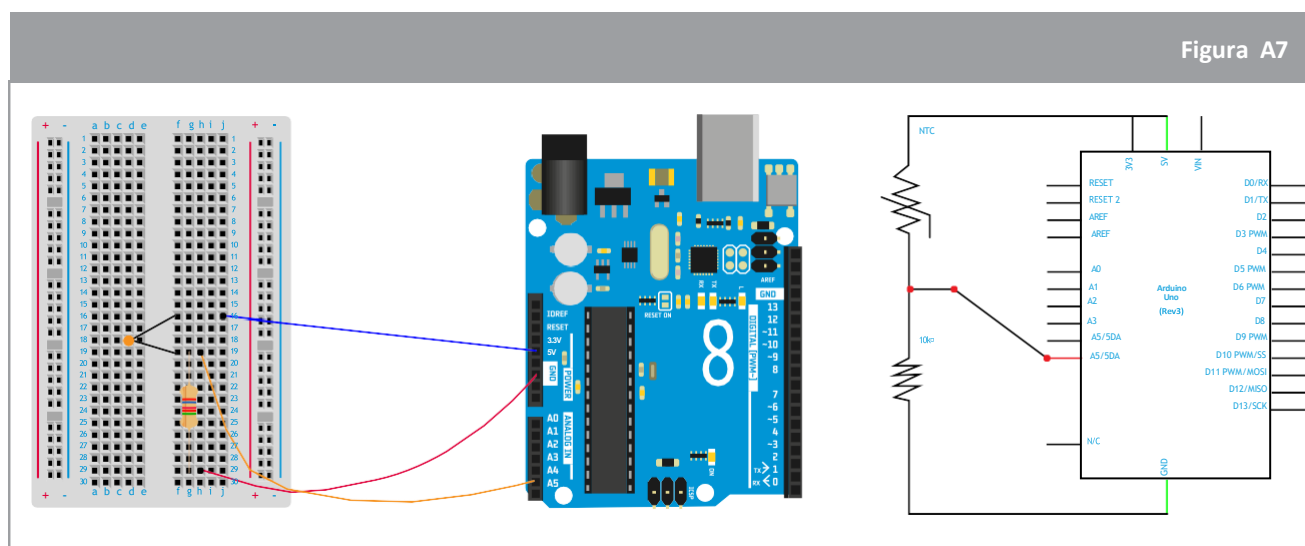
### Material necessário

- 1 Arduino Uno
- 1 breadboard
- 1 sensor de temperatura (termistor\*)
- 1 resistência de 10kΩ (castanho/preto/preto/vermelho/castanho)
- 3 fios

### Exercício

#### Montagem

- Ligue o termistor à breadboard como mostra a Figura A7.
- Ligue a resistência de 10kΩ (castanho/preto/preto/vermelho/castanho) ao termistor como mostra a Figura A7.
- Usando um fio, ligue o termistor ao pino de **5V** do Arduino Uno.
- Usando um fio, ligue a resistência ao pino **GND** (Terra) do Arduino Uno.
- Usando um fio, ligue o termistor e a resistência ao pino **A5** do Arduino.
- Para terminar ligue o Arduino UNO ao computador usando o cabo USB.



↑ Circuito do Arduino preparado para medir a temperatura

\* termistor: é uma resistência elétrica cuja resistência é grandemente diminuída por aquecimento

## Código

Agora os alunos vão escrever o seu próprio código. Uma lista de todos os comandos disponíveis pode ser encontrada fazendo uma pesquisa rápida no Google: comandos do Arduino. A Figura A8 mostra o exemplo de um código para medir a temperatura. Neste exemplo, a temperatura, que é lida a partir do sensor no circuito, chama-se **AnalogT**, e está em volts porque se trata de um sinal elétrico; não tem um significado físico de temperatura. A temperatura que está convertida em graus Celsius (a unidade de temperatura utilizada na Europa) chama-se **AnalogTf**. O nome do pino no Arduino Uno é A5. Assim que o código é carregado e começa a correr, abra a janela do Monitor em série\* para que a informação coletada seja visualizada.

Figura A8

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogT;
9   float AnalogTf;
10  AnalogT= float(analogRead(A5));
11  AnalogTf = (-40000/AnalogT) + 100;
12  Serial.println();
13  Serial.print("Temperature: ");
14  Serial.print(AnalogTf);
15  Serial.print("C");
16  delay(1000);
17
18 }
```

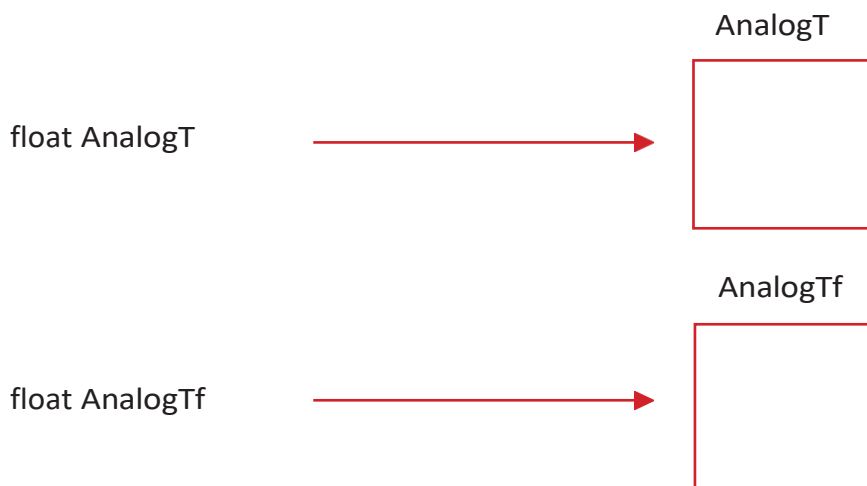
↑ Código do Arduino para medir a temperatura

\*ver a Tabela A1: comandos do Arduino.

Para explicações referentes ao void setup e ao void loop, por favor consulte a Atividade 1 Exercício 3.

**Linha 3:** À medida que a temperatura é impressa no ecrã do computador, é necessário predefinir a velocidade de dados (9600 bits/segundo) para a transmissão da informação entre o Arduino e o computador.

**Linhas 8-9:** Estas duas linhas do código são usadas para definir variáveis chamadas AnalogT e AnalogTf. Estes nomes são escolhidos pelo programador. Para modelarmos este processo vamos utilizar uma explicação gráfica. Quando escreve float AnalogT, dentro da memória do computador, cria uma caixa ou um espaço vazio chamado AnalogT que tem um tamanho específico indicado pelo 'float'. Float significa que o número dentro da caixa tem um valor decimal.



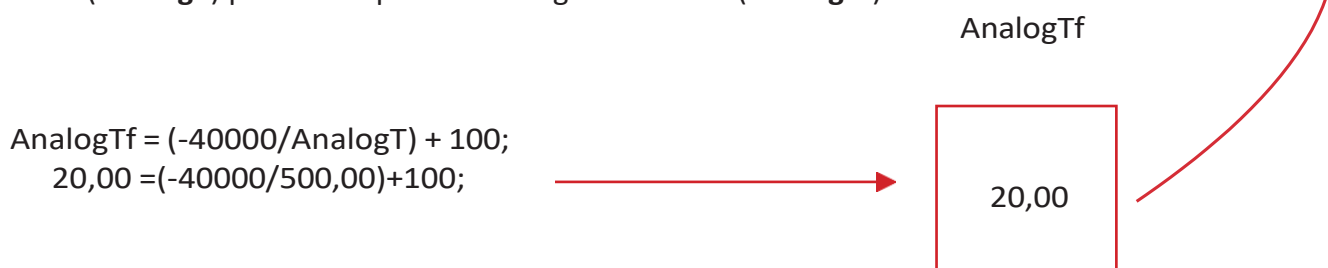
**Linha 10:** `AnalogT = float(analogRead(A5));`

O computador vai escrever o número lido pelo pino A5 no Arduino Uno (por exemplo 500,00) dentro da caixa **AnalogT**. Usamos a função **analogRead()** para ler qualquer pino do Arduino.



**Linha 11:** `AnalogTf = (-40000/AnalogT) + 100;`

Esta linha é a operação matemática que converte a temperatura em Volts (**AnalogT**) para a temperatura em graus Celsius (**AnalogTf**).



**Linhas 12-15:** Para mostrar no ecrã o valor da temperatura, tem de utilizar a função `print`. Esta função recupera o número dentro de qualquer caixa e coloca-o no Serial monitor( Monitor em série). Neste exemplo, a função vai imprimir o número que está dentro da caixa chamada **AnalogTf**. Todas as palavras que estão dentro de “...” são impressas diretamente no serial monitor.

**Linha 16:** A função de `delay` é usada para se esperar 1000 milissegundos para apresentar a temperatura no serial monitor.

## Discussão

Os alunos terão, provavelmente, de calibrar os seus sensores de temperatura. Eles podem utilizar outro termómetro para medir a temperatura real, como referência para a calibração do seu sensor. Para fazerem uma calibração aproximada, eles correm o código como está escrito e se o resultado da informação não for o mesmo que a temperatura de referência, tentam alterar o último número (+ 100) da linha 11 (AnalogTf) do código na Figura A8.

## Atividade

Pedir aos alunos para calibrarem o sensor de temperatura como se eles se encontrassem na superfície de Marte. Como se considera que a temperatura média da superfície Marciana é cerca de  $-60^{\circ}\text{C}$ , e de  $+14^{\circ}\text{C}$  para a Terra, os alunos deveriam obter um valor (por cálculo) de  $-74$ . O código deveria ser alterado para:

$\text{AnalogTf} = (-40000/\text{AnalogT}) + 100 - 74;$

## → Atividade 4: Medindo a pressão

Nesta atividade, os alunos medem a pressão ambiente com o Arduino. O objetivo é ler a pressão a partir do sensor (MPX4115A) e compreender como se escreve e utiliza o código para controlar o sensor. Pode encontrar mais informação fazendo uma pesquisa rápida no Google: Arduino MPX4115A.

### Material necessário

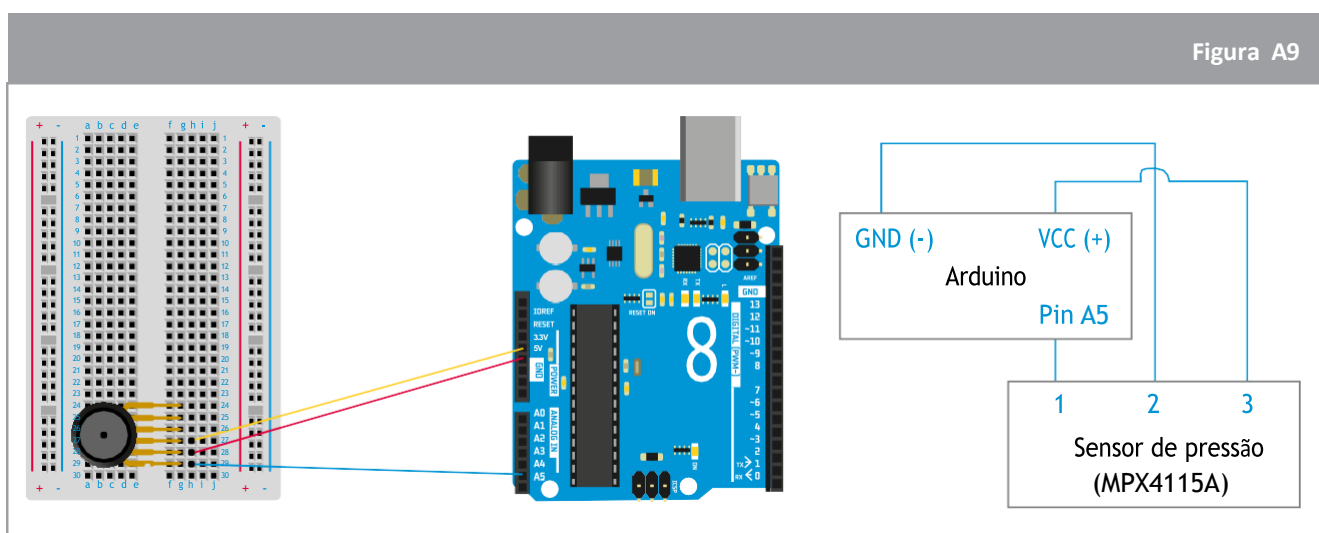
- 1 Arduino Uno
- 1 breadboard
- 1 sensor de pressão
- 3 fios

### Exercício

#### Montagem

- Ligue o sensor de pressão à breadboard.
- Usando um fio, ligue o terminal 1 do sensor de pressão (marcado por um entalhe) ao pino **A5**.
- Usando um fio, ligue o terminal 2 do sensor de pressão à linha **GND** (terra) do Arduino.
- Usando um fio, ligue o terminal 3 do sensor de pressão ao pino **5V** do Arduino.
- Para terminar, ligue o Arduino UNO ao computador usando o cabo USB.

Figura A9



↑ Circuito do Arduino preparado para medir a pressão



## Código

Neste caso, os alunos utilizam os mesmos comandos da atividade anterior sobre temperatura; Mais uma vez vão focar-se na calibração do sensor. Um passo importante e muito útil quando se usam sensores, é o de consultar as respetivas fichas técnicas, que contêm informação detalhada sobre o funcionamento do sensor. Neste caso, podemos encontrar, facilmente, informação útil fazendo uma rápida busca no Google: Ficha técnica MPX4115A.

O sensor de pressão converte os valores medidos da pressão numa voltagem. Trata-se de um sinal analógico que é convertido num sinal digital. Para converter as voltagens medidas para valores da pressão ambiente (em unidades pascals, Pa, do Sistema Internacional de Unidades), é necessário usar a função transfer do sensor. Esta função estabelece a relação matemática entre os valores de voltagem medidos pelo sensor e os valores equivalentes da pressão. Esta função pode ser encontrada na ficha técnica do sensor e pode variar de sensor para sensor.

Se consultarmos a ficha técnica, vamos encontrar a seguinte função transfer:  $V_{out} = V_s (P \times 0,009 - 0,095)$ . Só precisamos de isolar o valor de P (Pressão) e considerar que  $V_s = 1024$  (a voltagem lida está dividida em 1024 etapas).  $V_{out}$  = valor analógico que vamos ler no pino A5.

Em baixo está um exemplo do código que imprime os valores da pressão. Utilize-o como referência.

Figura A10

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogP;
9   float AnalogPf;
10  AnalogP= float(analogRead(A5));
11  AnalogPf = ((AnalogP/1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure: ");
14  Serial.print(AnalogPf);
15  Serial.print("kPa");
16  delay(1000);
17
18 }
```

↑ Código do Arduino para medir a pressão

O nome que se dá à pressão em volts é AnalogP e para a pressão em pascal é AnalogPf. O nome do pino onde se encontra ligado o sensor de pressão é A9.

Para calibrarem os seus sensores de pressão os alunos podem verificar qual a pressão correspondente à sua localização, na internet. Para fazerem uma calibração aproximada, eles correm o código como se encontra escrito e se a informação não for igual à pressão de referência, podem alterar a linha 11 (AnalogPf) do código.

## → Atividade 5: Medindo a altitude

A pressão varia com a altitude, por isso a pressão pode ser utilizada para medir a altitude. Para medir a altitude, os alunos têm de acrescentar esta parte do código ao código que mede a pressão e que se encontra dentro do loop.

Em baixo encontra-se um exemplo do código que imprime a altitude em metros. Use-o como referência.

Figura A11

```
float Altitude;
float Altitudef;
  Altitude = pow(AnalogPf/101.325,0.1903);
  Altitudef = (1-Altitude)*44300 + 100;
Serial.print("Altitude: ");
Serial.print(Altitudef); Serial.print ("meters");
```

↑ [Código do Arduino para medir a pressão](#)

O nome dado à altitude em metros é Altitudef. Neste exemplo a pressão em unidades de kPa tem o nome de AnalogPf.

A partir das sugestões dadas na caixa “Sabias que”, os alunos devem compreender que à medida que a altitude aumenta, a temperatura da atmosfera diminui, e esta relação deve aparecer no seu gráfico da altitude e da temperatura.

## Como utilizar o Arduino na sala de aula

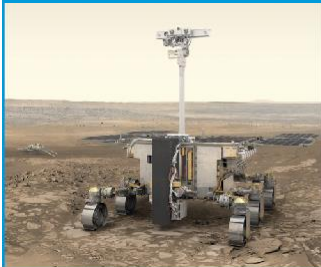
O Arduino é uma ferramenta excelente para introduzir aos alunos a programação básica e a ampla variedade de aplicações que contem. Também pode ser usado para ensinar e aprender matérias relacionadas com STEM (ciência, tecnologia, engenharia e matemática). Alguns exemplos são:

- **Introdução aos fundamentos dos circuitos** – A Atividade 1 pode ser utilizada para introduzir aos alunos circuitos simples.
- **Introdução à análise de dados** – Os alunos recolhem dados do mundo real (temperatura e/ou pressão) e analisam esses dados utilizando gráficos. Seguidamente leem a informação e tiram conclusões.
- **Introdução às variáveis do tempo e ao modo como a informação é recolhida** – Os alunos podem criar a sua própria estação meteorológica e preverem dias com sol e dias nublados.

# → CONHECE O ARDUINO!

## Introdução à programação do Arduino utilizando C++

### Sabias que?



A missão do programa ExoMars 2020 vai colocar um rover Europeu e material científico Russo na superfície de Marte para perceber se alguma vez existiu aí vida!

### → Atividade 0: Vamos começar

Todos nós utilizamos tecnologia todos os dias! Nas missões espaciais, os engenheiros utilizam a tecnologia para controlar e comunicar com robots de modo a aumentar o nosso conhecimento científico. Nesta atividade, vais ser o engenheiro que vai investigar Marte, o Planeta Vermelho, e para isso vais utilizar uma ferramenta tecnológica chamada Arduino. Vamos descobrir como identificar os componentes utilizados no Arduino!

### Material necessário

- O Kit básico do Arduino

### Componentes da Estação em Terra:

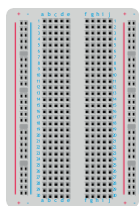
1. Trabalha aos pares. Abre o teu kit do Arduino e associa os componentes à descrição da página seguinte

Figura A1

<p>a. breadboard</p>	<p>b. Sensor de pressão</p>	<p>c. Resistências</p>	<p>d. Cabo USB</p>
<p>e. Fios(cabos)</p>	<p>f. LEDs (Diodo emissor de luz)</p>	<p>g. Sensores de temperatura</p>	<p>h. Placa do Arduino</p>

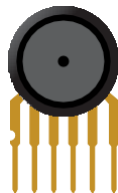
COMPONENTE

DESCRIÇÃO



breadboard

Tem uma forma circular e mede a pressão ambiente



Sensor de pressão

Tem pinos de entrada e de saída e funciona como um computador



Resistências

É um suporte branco utilizado para ligar, com facilidade, componentes eletrônicos



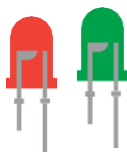
Cabo USB

Tem dois terminais compridos que estão ligados a uma “cabeça” azul e mede a temperatura



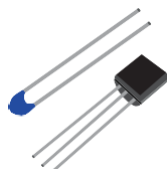
Fios (cabos)

Têm forma cilíndrica com faixas coloridas e diminuem o fluxo da corrente num circuito



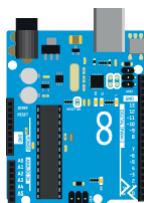
LEDs (Diodo emissor de luz)

São vermelhos e verdes e emitem luz quando percorridos pela corrente elétrica



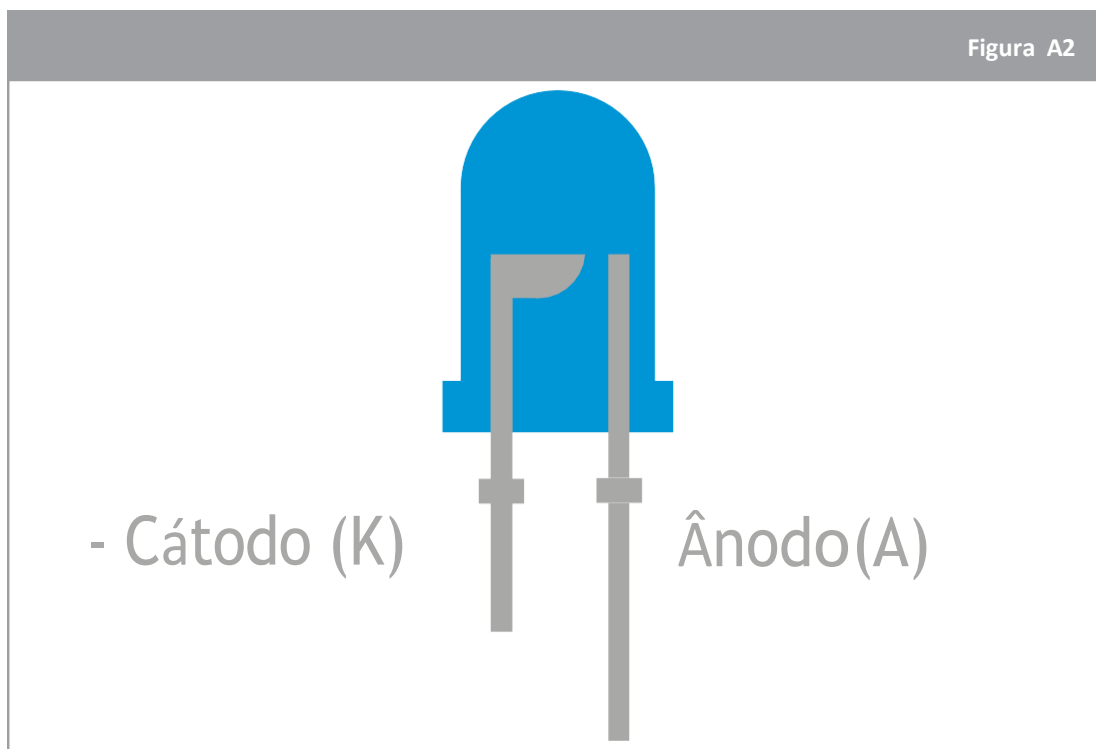
Sensores de temperatura

Têm várias cores e são usados para ligar componentes e conduzir a corrente elétrica

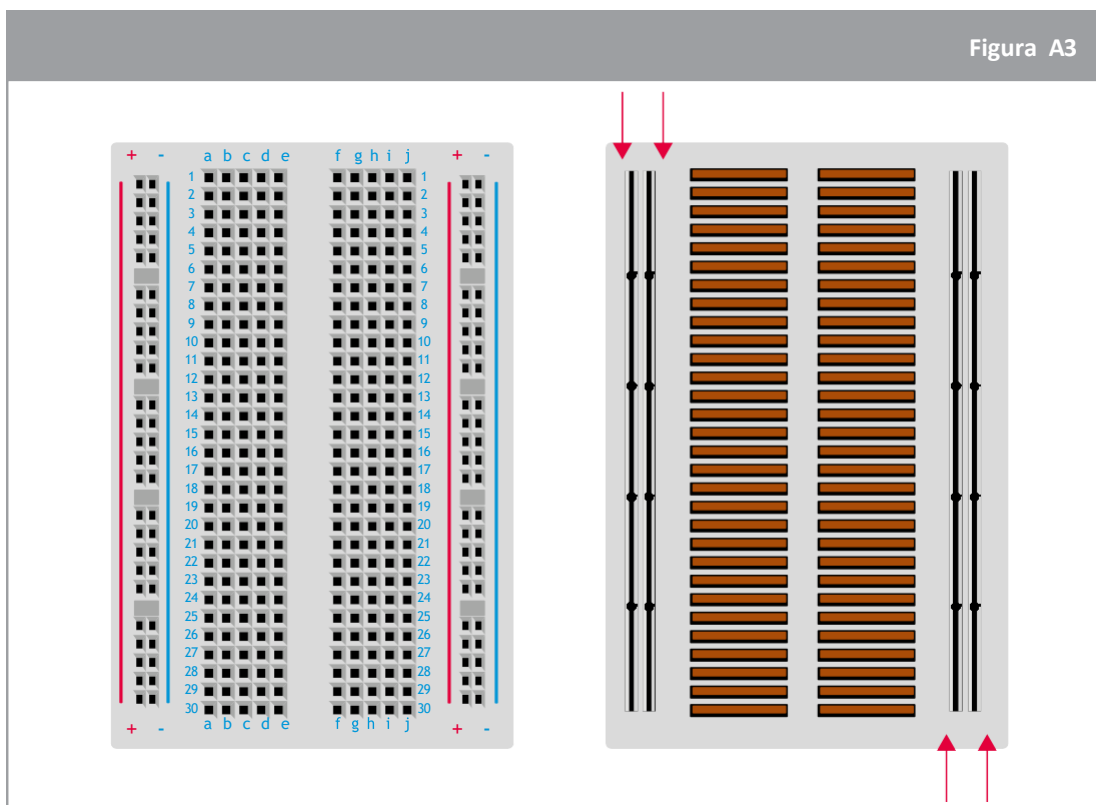


Placa Arduino

Um cabo preto e comprido usado para ligar o Arduino Uno ao computador



↑ LED: Observa que o LED tem um terminal comprido e outro curto. O terminal comprido chama-se ânodo e o curto cátodo.



↑ Breadboard: Observa que todos os pinos de cada fila estão ligados.

## → Atividade 1: Faz-me piscar!

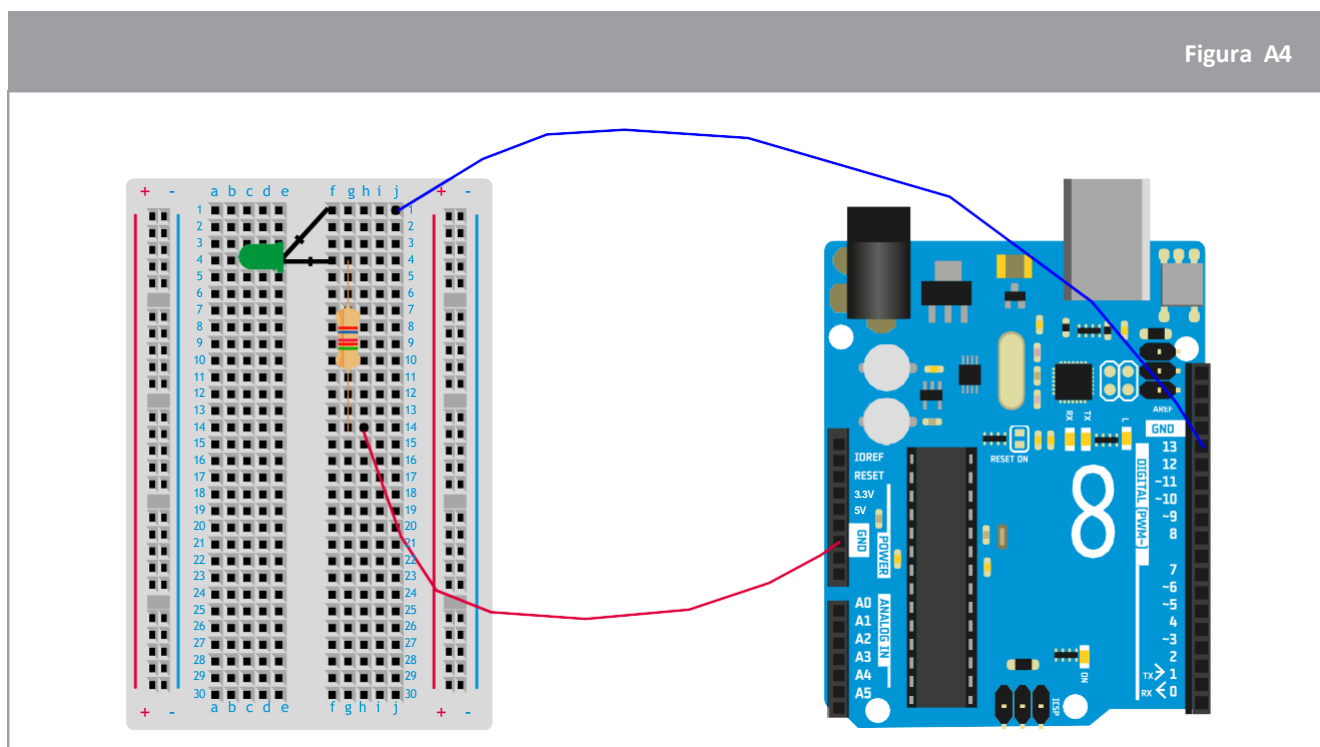
Para controlar objetos interativos, tens de usar uma ferramenta tal como o Arduino, constituída por vários componentes eletrónicos. Como és o engenheiro encarregado de estudar o Planeta Vermelho, precisas de comunicar com o rover para coordenar as suas experiências e recolher os dados. Nesta atividade vais aprender a linguagem de programação C++ para comunicares com o Arduino Uno, e assim aprender a controlar a luz!

### Material necessário

- 1 Arduino Uno
- 1 breadboard
- 1 LED verde
- 1 resistência de 220Ω (vermelho/vermelho/preto/preto/castanho)
- 2 fios

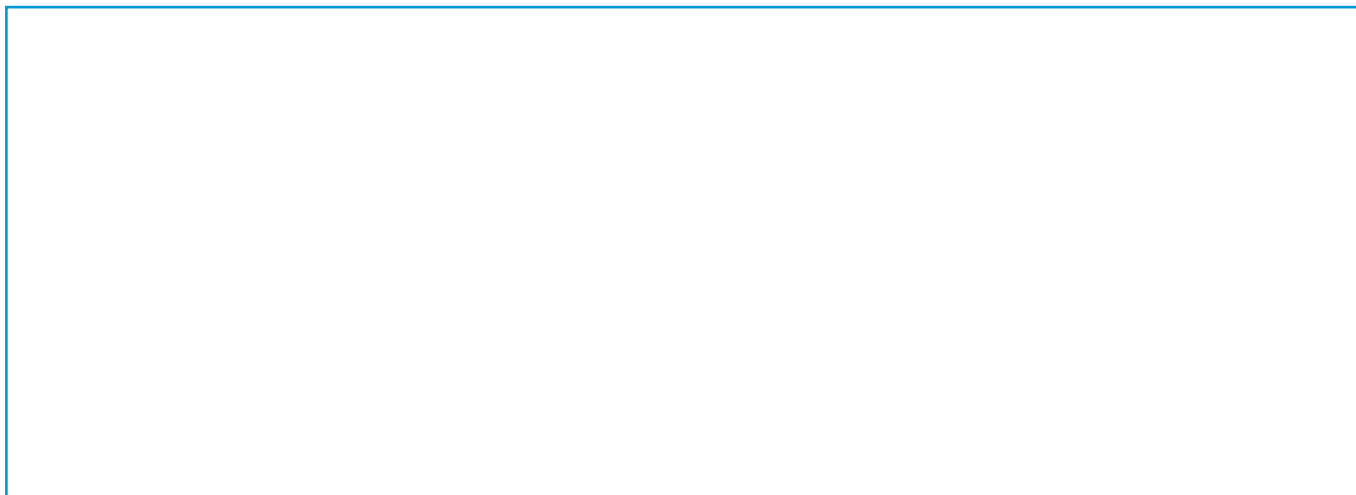
### Exercício

1. Segue estas instruções cuidadosamente para montares a breadboard e controlares o LED verde:
  - a. Liga o LED verde na breadboard como mostra a Figura A4.
  - b. Liga a resistência de 220Ω (vermelho/vermelho/preto/preto/castanho) ao terminal mais curto do LED como se mostra na Figura A4.
  - c. Usando um fio, liga o terminal mais comprido do LED ao pino número **13** do Arduino Uno.
  - d. Usando um fio, liga a resistência ao pino **GND**(Terra) do Arduino Uno.
  - e. Liga o Arduino Uno ao computador, utilizando o cabo USB.



↑ O circuito do Arduino preoarado para fazer piscar o LED verde

2. Na caixa em baixo desenha o circuito elétrico correspondente às ligações elétricas da Figura A4. Utiliza estes símbolos:



3. Agora que a breadboard está pronta, tens de enviar instruções ao Arduino Uno para fazer piscar o LED.

Abre o software do Arduino (IDE), no computador, e clica no ficheiro → Examples → Basics → Blink.

O código de piscar vai aparecer no ecrã e está apresentado na figura em baixo. Lê atentamente a explicação para a compreenderes na totalidade:

### Upload the program to the Arduino

```

1  /*
2   * Blink
3   * Turns on an LED on for one second, then off for one second, repeatedly.
4   *
5   * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6   * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7   * of use the correct LED pin whatever is the board used.
8   * If you want to know what pin the on-board LED is connected to on your Arduino model, check
9   * the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10  *
11  * This example code is in the public domain.
12  *
13  * modified 8 May 2014
14  * by Scott Fitzgerald
15  *
16  * modified 2 Sep 2016
17  * by Arturo Guadalupi
18  */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
35

```

The grey text is comments, it explains what the code does

LED\_BUILTIN is the name/number of the pin where the LED is connected


1000ms is the delay between turning the LED on

HIGH = LED is on  
LOW = LED is off

- a. Antes de carregares o código para o Arduino Uno, tens de emparelhar o nome do pino no código com o número do pino onde o LED verde está ligado ao Arduino Uno. Explica o que mudaste no código para controlares o LED verde:

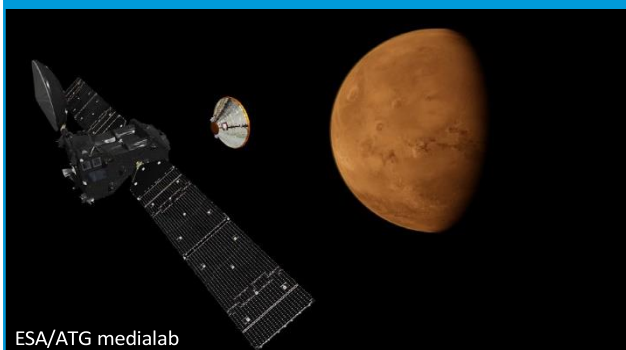
---

---

- b. Carrega o código para o Arduino Uno clicando em , no canto superior esquerdo do ecrã.

Parabéns, o LED verde está a piscar!

### Sabias que?



Marte está tão longe que os sinais demoram bastante a viajar da nave espacial ou do rover até à Terra. Este atraso na comunicação torna desafiante ter uma conversa com o rover, ou reagir rapidamente se alguma coisa inesperada acontecer.



## → Atividade 2: Pisca um S.O.S.!

Agora que já sabes como fazer piscar um LED utilizando o código do Arduino, vais aprender, com esta atividade, a comunicar usando um LED e a enviar uma mensagem para um rover em Marte. Na verdade está previsto a passagem de uma muito forte tempestade de areia pelo rover! Para prevenir danos no rover, faz com que ele envie um S.O.S.!

### Sabias que?



NASA/JPL-Caltech/MSSS

Esta imagem de uma tempestade de pó em Marte mostra as nuvens que resultam de alterações da pressão atmosférica, da temperatura e da altitude, por causa de deslocamentos verticais, tais como os que ocorrem quando o vento sopra sobre uma montanha ou sobre a parede de uma cratera.

### Material necessário

- Circuito construído na Atividade1

### Exercício

Usa o código Morse para fazer piscar o LED verde. Na Figura A5, um ponto representa um sinal curto, que vai acender o LED durante pouco tempo, e um traço significa um sinal longo, que vai acender o LED por um período de tempo maior. O código para enviar um S é dado na Figura A6.

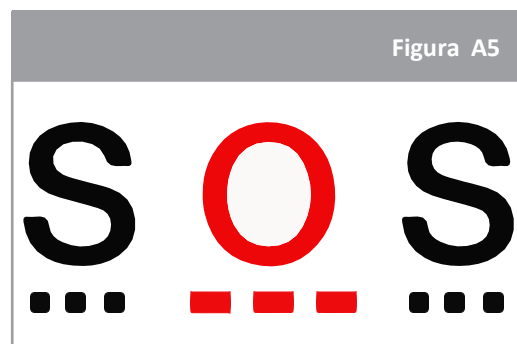


Figura A5

↑ SOS em código Morse

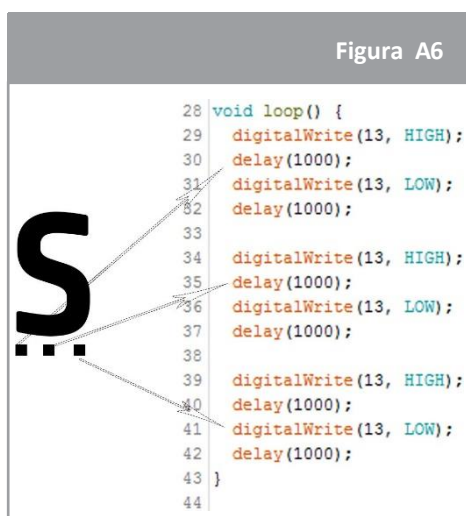


Figura A6

↑ Código para enviar um S em Morse

explica o que precisas alterar no código para enviar um O:

---



---



---



---

Altera o código para enviases um S.O.S., depois carrega-o no Arduino e testa-o!

## → Atividade 3: Medindo a temperatura

A tecnologia do Arduino permite que controles um LED ligado ao Arduino Uno, utilizando um código escrito num computador portátil. Quando carregas o código, ele envia as instruções para o Arduino Uno ativando os componentes eletrônicos. Na missão ExoMars, vai ser usada tecnologia para investigar o meio ambiente de Marte, através de alguns sensores. Nesta atividade, vais descobrir como medir a temperatura da sala de aula e simular a temperatura em Marte!

### Material necessário

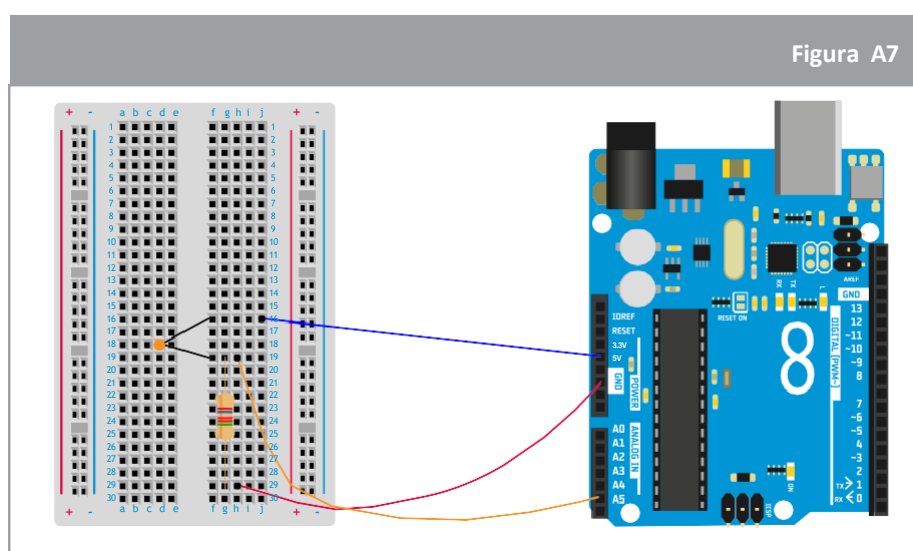
- 1 Arduino Uno
- 1 breadboard
- 1 sensor de temperatura (termistor)
- 1 resistência de 10kΩ (castanho/preto/preto/vermelho/castanho)
- 3 fios

### Exercício

1. Para medir a temperatura da sala de aula, tens de construir o circuito, mostrado em baixo, na breadboard

Segue atentamente estas instruções:

- a. Liga o termistor na breadboard como se mostra na Figura A7.
- b. Liga a resistência de 10kΩ (castanho/preto/preto/vermelho/castanho) ao termistor como se mostra na Figura A7.
- c. Utilizando um fio, liga o termistor ao pino de **5V** do Arduino Uno.
- d. Utilizando um fio, liga a resistência ao pino **GND** (Terra) do Arduino Uno.
- e. Utilizando um fio, liga o termistor e a resistência ao pino **A5** do Arduino.
- f. Liga o Arduino Uno ao computador utilizando um cabo USB.



↑ Circuito do Arduino preparado para medir temperaturas

2. A figura A8 mostra um código, parcialmente incompleto, para medir a temperatura. Três valores, apresentadas como caixas nas cores (vermelha, verde e azul), têm de ser preenchidas para darem instruções corretas ao Arduino Uno de modo a que ele meça a temperatura. Vamos descobrir como completar este código.

Figura A8

```

1
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7
8   float ;
9   float ;
10
11    = float(analogRead());
12    = (-40000/) + 100;
13
14   Serial.println();
15   Serial.print("Temperature in classroom: ");
16   Serial.print();
17   Serial.print("C");
18   delay(1000);
19
20 }

```

↑ Código do Arduino para medir a temperatura

- a. Qual é a unidade utilizada para medir a temperatura na Europa?

O sensor de temperatura ligado ao Arduino Uno vai, primeiro, ler a temperatura noutra unidade, o volt. Para converter este número de volts para graus Celsius, vais utilizar a fórmula escrita na linha 12 da Figura A8. Por fim, vais apresentar a temperatura no ecrã utilizando a função Serial.print do código.


- b. Define a variável que corresponde à temperatura em volts:

Define a variável que corresponde à temperatura em graus Celsius:

Procura o pino do Arduino onde está ligado o sensor de temperatura:

- c. Agora estás pronto para escreveres o código no teu computador portátil. Abre o software do Arduino e escreve o código da Figura A8, substituindo as caixas pelas variáveis que definiste para as cores vermelha, verde e azul. Tem cuidado de modo a substituíres todas as caixas coloridas pelas respetivas variáveis.

Carrega o código para o Arduino Uno clicando em 

- d. Para veres as medições da temperatura, tens de abrir uma janela específica no software do Arduino chamada Serial monitor. Procura este símbolo  na barra das ferramentas para lhe acederes.

## Discussão

1. Discute a relevância das medições da temperatura com os teus colegas. Como podes verificar que o sensor de temperatura mede com precisão a temperatura da sala de aula?

---

---

---

---

2. Qual é a linha do código, na Figura A8, que podes alterar para calibrar o sensor?

---

---

3. Para treinares um pouco mais com o Arduino, calibra o sensor de temperatura como se ele estivesse na superfície de Marte. Considera que a temperatura média na superfície de Marte é cerca de  $-60^{\circ}\text{C}$ .

a. Temperatura média na Terra =

---

b. Escreve a nova fórmula de calibração do sensor de temperatura aqui:

---

## → Atividade 4: medindo a pressão

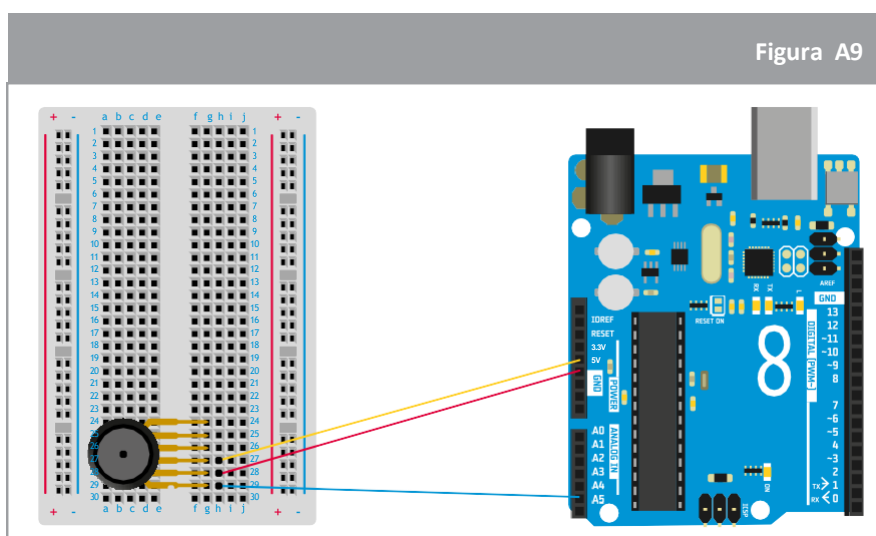
Nesta atividade, vais aprender a medir a pressão atmosférica utilizando o Arduino e um sensor de pressão.

### Material necessário

- 1 Arduino Uno
- 1 breadboard
- 1 sensor de pressão
- 3 fios

### Exercício

1. Para medir a pressão na sala de aula, tens de começar por construir o circuito na breadboard (Figura A9). Segue atentamente estas instruções (Nota: Cada pino no sensor de pressão tem um comportamento diferente, por isso certifica-te que a orientação é a mesma que a apresentada na figura):



↑ Circuito do Arduino preparado para medir a temperatura

2. A figura A10 mostra um código, parcialmente incompleto, para medir a pressão. Três variáveis, apresentadas como caixas nas cores (vermelha, verde e azul), têm de ser preenchidas para dar instruções corretas ao Arduino Uno de modo a que este meça a pressão. Vamos descobrir como completar este código.

Figura A10

```

1 void setup() {
2
3   Serial.begin(9600);
4 }
5 void loop() {
6
7   float [caixa vermelha] ;
8   float [caixa verde] ;
9
10  [caixa vermelha] = float(analogRead([caixa azul]));
11  [caixa verde] = (([caixa vermelha] /1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure in classroom: ");
14  Serial.print([caixa verde]);
15  Serial.print("kPa");
16  delay(1000);
17
18 }

```

↑ Código do Arduino para medir a pressão

- a. Qual é a unidade utilizada para a pressão no S.I. (Sistema Internacional de Unidades)?

- b. Define a variável que corresponde à pressão em volts:

- c. Define a variável que corresponde à pressão kilopascal:


Procura o pino do Arduino onde está ligado o sensor de pressão:


O sensor de pressão ligado ao Arduino Uno vai ler em primeiro lugar a pressão noutra unidade, o volt, que é uma unidade de electricidade. Para converter este número de volts para pascals, usamos esta fórmula da datasheet do sensor  $V_{out} = V_s (P \times 0,009 - 0,095)$ .

Finalmente, apresenta as medições da pressão no ecrã utilizando a função Serial.print no código.

- d. Usando a expressão  $V_{out} = V_s (P \times 0,009 - 0,095)$ ; calcula a fórmula de calibração para completares a linha na Figura A10. Como a voltagem lida está dividida em 1024 etapas, considera  $V_s = 1024$ .

- e. Agora estás preparado para escrever o código no teu computador portátil. Abre o software do Arduino e escreve o código da Figura A10, completando-o com as variáveis que definiste para as caixas vermelhas, verdes e azuis. Tem cuidado de modo a substituíres todas as caixas coloridas pelos respetivos valores!

Carrega o código para o Arduino Uno clicando em 

- f. Para ver as medições da pressão, tens de abrir uma janela específica no software do Arduino chamada Serial monitor. Procura este símbolo  na barra das ferramentas para lhe acederes:

## Discussão

Discute a relevância das medições da pressão com os teus colegas. Como podes verificar que o sensor de pressão mede com precisão a pressão atmosférica da sala de aula?

---

---

---

---

---

---

Qual é a linha do código da Figura A10 que tens de alterar para calibrar o sensor?

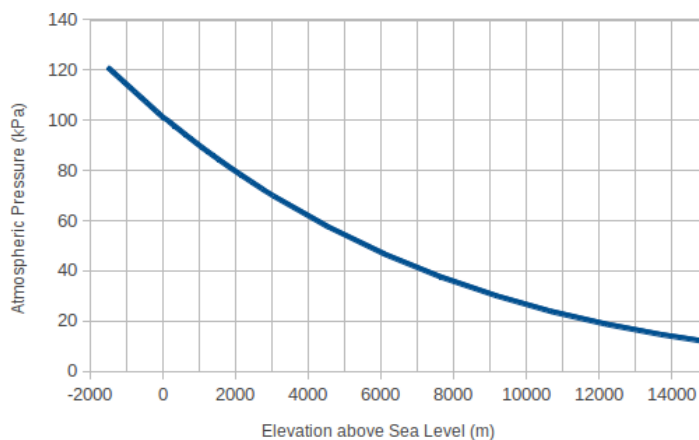
---

---

---

## → Atividade 5: medindo a altitude

A pressão do ar varia com a altitude. Por exemplo, a pressão do ar junto ao mar é superior à pressão do ar nas montanhas. Podes ver com varia consultando o gráfico. Com um pouco de matemática e algum conhecimento da física que está na base das leis dos gases, podemos calcular exatamente qual é a relação entre a pressão e a altitude. Por agora, não temos de nos preocupar com isto. Nesta atividade vamos utilizar as tuas medições da pressão para determinar a que altitude te encontras, utilizando uma versão simplificada da fórmula que dá a relação pressão-altitude e que se chama fórmula barométrica.



## Material necessário

- O circuito do Arduino construído na Atividade 4
- O código para medir a pressão utilizado na Atividade 4

## Exercício

1. Abre o código da pressão que escreveste na Atividade 4 no software do Arduino. Acrescenta este código parcialmente incompleto dentro do void loop:

Figura A11

```

1 float [ ];
2 float [ ];
3
4 [ ] = pow([ ] / 101.325, 0.1903);
5 [ ] = (1 - [ ]) * 44300 + 100;
6
7 Serial.println();
8 Serial.print("Altitude in classroom: ");
9 Serial.print([ ]);
10 Serial.print("meters");
11 delay(1000);

```

↑ Código do Arduino para medir a altitude

2. Para determinar a altitude, vais utilizar as medições do sensor da pressão na fórmula escrita na linha 4 da Figura A11. Como percebeste nos exercícios anteriores, um sensor ligado ao Arduino Uno vai ler primeiro os valores das medições em volts. Para converter este número de volts para metros, vais utilizar a fórmula escrita na linha 5 da Figura A11. Por fim, a informação sobre a altitude pode ser apresentada usando a função Serial.print do código. Se a leitura não estiver calibrada pode ser ajustada alterando a linha 5 do código. .



3. a. Define a variável que corresponde à altitude em volts:

Define a variável que corresponde à altitude em metros:

Procura o nome que deste à caixa verde que corresponde à pressão em pascal (vê a Atividade 4):

b. Completa o código com as variáveis que definiste para as caixas amarela, roxa e verde. Tem cuidado para substituíres todas as caixas coloridas pelos seus respetivos valores! Seguidamente carrega o código.

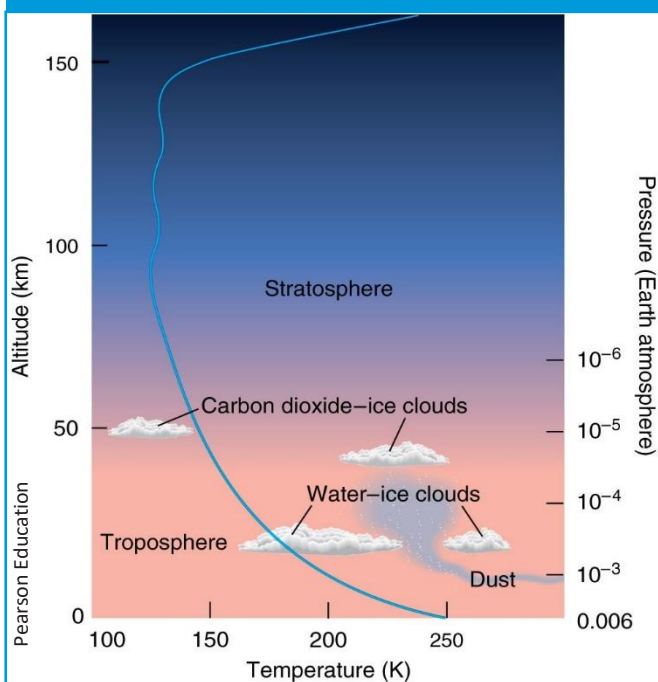
c. Para apresentar os dados da altitude, tens de abrir uma janela específica no software do Arduino chamada Serial monitor.

4. Combina ambos os códigos da temperatura e da altitude num sketch único e carrega-o no Arduino Uno.

Recolhe as informações da temperatura e da altitude desde o chão até ao teto da tua sala de aula e depois faz o gráfico com os resultados.

Consegues explicar a forma do gráfico?

### Sabias que?



A atmosfera Marciana é constituída por uma camada extremamente fina de gás, principalmente dióxido de carbono, que se estende desde a superfície de Marte até ao início do espaço. O Sol aquece a superfície de Marte e algum deste calor aquece o gás junto à superfície. O gás aquecido espalha-se ou sofre uma convecção através da atmosfera. Portanto, a temperatura do gás é mais elevada junto à superfície e diminui à medida que a altitude aumenta.

## → LINKS

Blogue do Arduino, é útil para encontrar as informações mais atuais sobre o Arduino: <https://blog.arduino.cc/>

Guias para utilizar os diferentes componentes do Arduino:  
<https://quarkstream.wordpress.com/>

Muitos projetos baseados no Arduino que podes tentar em casa encontram-se nos Instructables:  
<http://www.instructables.com/howto/arduino/>

HackADay contem muitos outros projetos do Arduino:  
<https://hackaday.io/list/3611-arduino-projects>