

PROGRAMAÇÃO E ROBÓTICA PARA UM ROBÔ *ARDUINO*

10, 11 E 12 NOVEMBRO 2022

AzoresBot

Quinta-feira

AzoresBot – Robô e competição

O que é um robô?

Como programar um robô?

Sexta-feira

Sensores e atuadores

Como funcionam os sensores?

Sábado

Vamos pôr em prática o que aprendemos

AzoresBot

Primeiro festival de robótica dos Açores;

<https://azoresbot2019.uac.pt/>

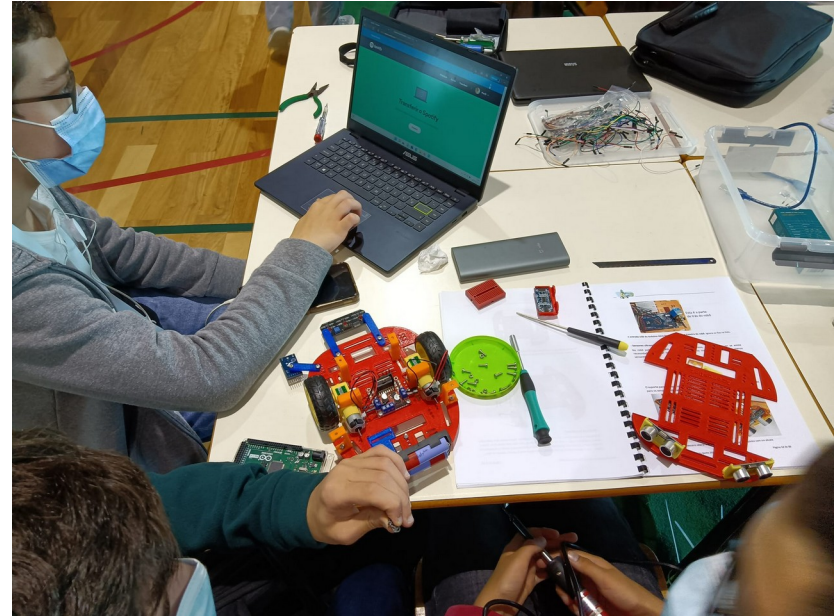
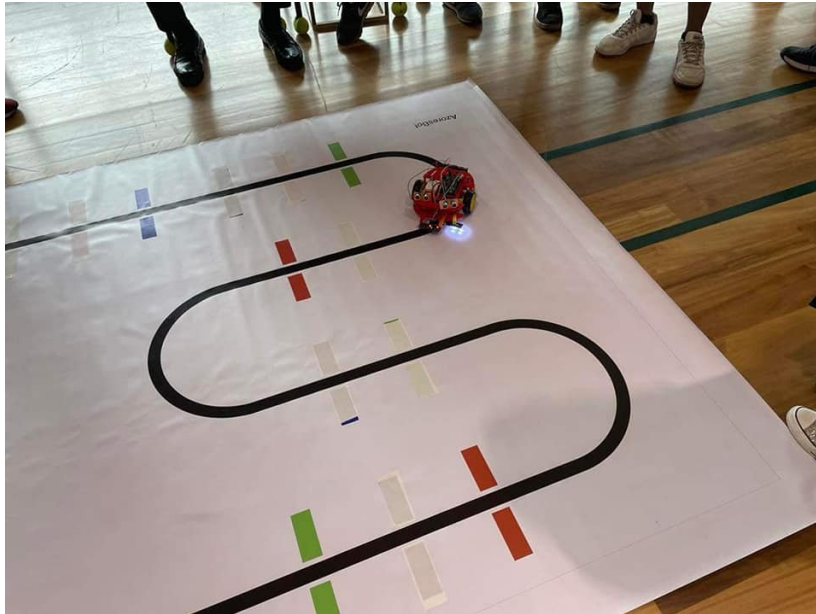
Primeiro robô desenhado e desenvolvido na região;

<https://gria.uac.pt/projects/azoresbot>

Trabalho colaborativo de professores universitários, de escolas profissionais e de escolas básicas e secundárias públicas de S.Miguel;

AzoresBot

Segunda edição do AzoresBot - 2022
<https://azoresbot2022.uac.pt/>



Novo desafio - Vulcano



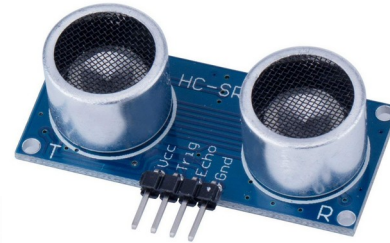
O que é um robô?

“Um robô é uma máquina autónoma capaz de sentir o seu ambiente, realizar cálculos para tomar decisões e realizar ações no mundo real.”

Retirado de <https://robots.ieee.org/learn/what-is-a-robot/>

AzoresBot

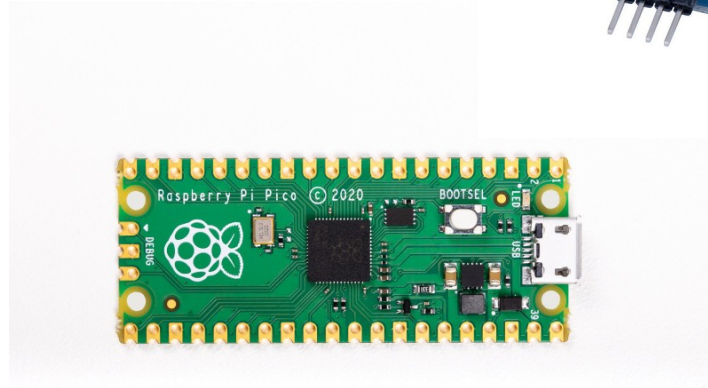
Sensores



Atuadores



Computador



Microcontrolador

Microcontrolador é um pequeno computador (SoC - System on a Chip) num único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída.



Arduino

Plataforma de prototipagem eletrónica de hardware livre e de placa única;

Microcontrolador Atmel AVR (Atmega328p);

Processador de 8 bits;

ESP32

Xtensa dual-core 32-bit LX6 microprocessador - 240 MHz;
Até 44 GPIO (General Purpose Input-Output) programáveis;
WiFi 2.4 GHz (IEEE 802.11 b/g/n);
Bluetooth 5 (LE);

<https://en.wikipedia.org/wiki/ESP32>

Alimentação elétrica Arduino e ESP32

Arduino

Jack – 7V – 12V

USB – 5V

Pino Vin – 7V - 12V

Pino 5V – 5V

Portas lógicas - 5V

ESP32

USB – 5V

Pino 5V – 5V

Pino 3.3V – 3.3V

Portas Lógicas – 3.3V

Limite de corrente (em amperes)

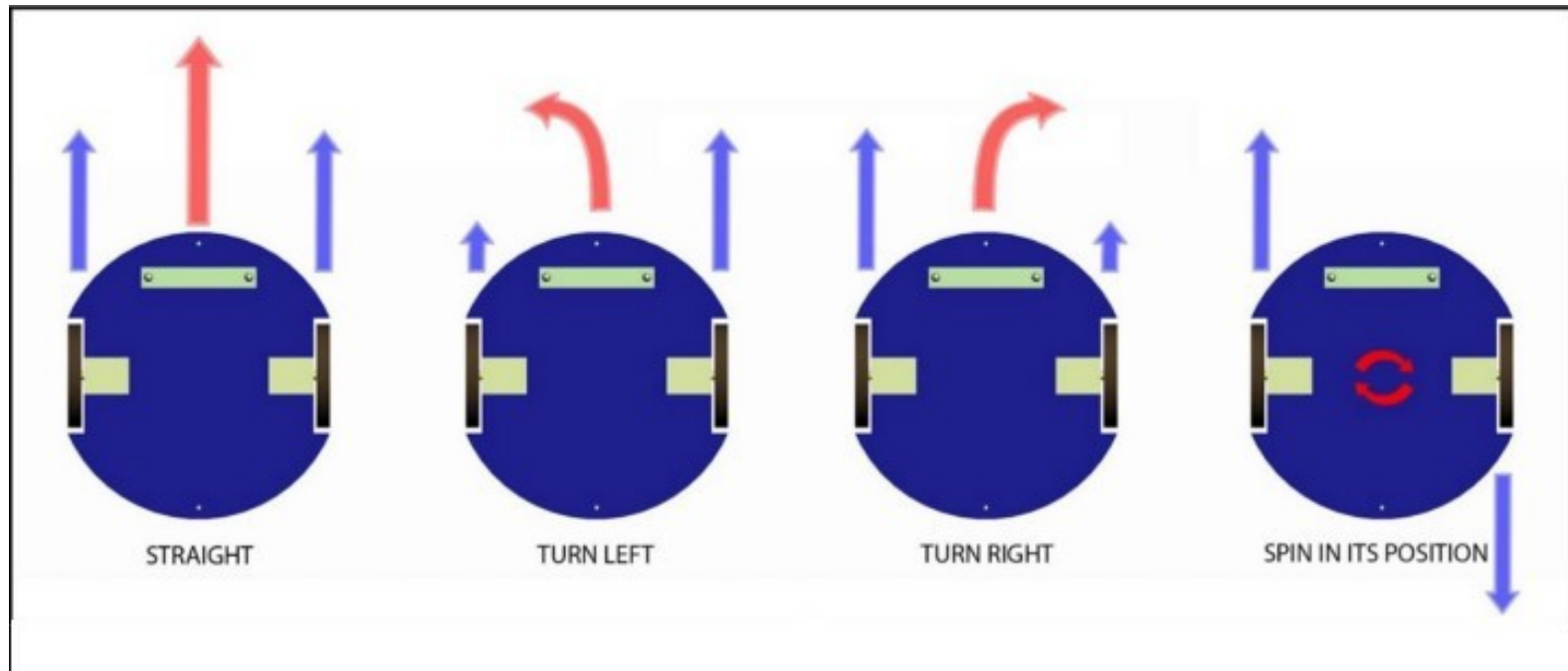
Arduino - ESP32

DC Current per I/O Pin - **40.0mA**

DC Current VCC and GND Pins - **200.0mA"**



Robô diferencial



AzoresBot

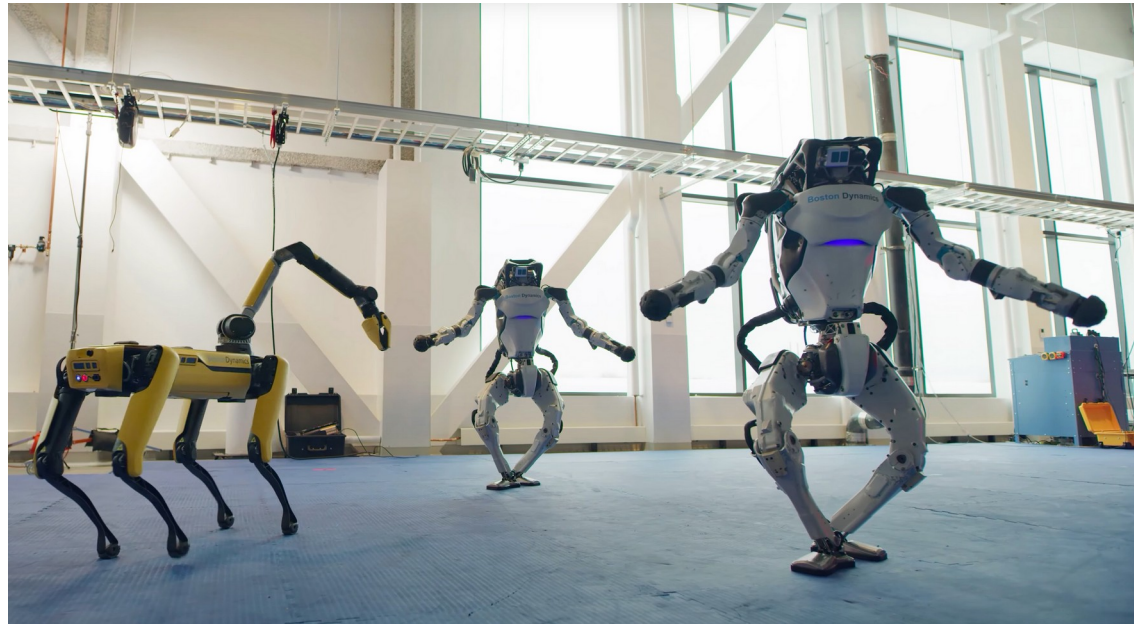
Robot omnidirecional



<https://www.youtube.com/watch?v=DHVPd3MUEis>

AzoresBot

Robôs com pernas



https://www.youtube.com/watch?v=UAG_FBZJVJ8

Como programar?

Linguagem de programação:

A linguagem de programação usada nas plataformas Arduino é a linguagem C/C++;

A linguagem C/C++ é uma linguagem de alto nível e necessita ser convertida em linguagem máquina usando um compilador;

Ao abrir o IDE pela primeira vez encontramos um conjunto de código já pronto e que corresponde a duas funções principais na programação do Arduino. Esta é a estrutura de qualquer programa para Arduino (também designada sketch).

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Na função **void setup()** são definidas algumas das configurações iniciais do programa. Esta função é executada apenas uma vez.

A função **void loop()** é a função principal do programa e será executada indefinidamente.

Nestas funções o código deve ser sempre inserido entre as duas chavetas **{ ... }**; **void** indica que estas funções não retornam nenhum valor.

Um programa para a plataforma Arduino pode ser dividido nas seguintes partes:

- **Variáveis**
- **Funções**
- **Estruturas**

Variáveis

Servem para armazenar dados num programa;

Os dados podem ser de diferentes tipos, como por exemplo:

int (número inteiro);

boolean (valor verdadeiro - true - ou falso - false);

byte (um byte ou sequência de oito bits);

char (um caracter);

string (sequência de caracteres);

void (tipo vazio);

Para usar uma variável precisamos de especificar o tipo de dado que a variável irá possuir e de atribuir um identificador - sequencia de uma ou mais letras, números ou underscore;

int x

char uma_letra

boolean sensor

**Quando declaramos uma variável o seu valor é indeterminado.
Quando lhe atribuímos um valor estamos a inicializar a variável;**

```
int x = 56;
```

```
char uma_letra ='k'
```

```
boolean sensor = true
```

Constantes são idênticas às variáveis mas como o nome indica não variam durante a execução do programa. Podemos declará-las das seguintes formas:

```
const int x = 56;
```

```
#define x 56
```

Funções

- Sequência de instruções que pode ser reutilizada ao longo de um programa;
- Sintaxe:

tipo nome da função (parametro1, parametro2, ...){ instruções }

```
int soma (int a, int b){
```

```
int resultado = a + b;
```

```
return resultado;
```

```
}
```

Instruções de controlo de execução:

- **Decisão binária if-else;**
- **Decisão múltipla switch;**
- **Repetição condicional while e for.**

AzoresBot

```
if(x==10){  
    x=0;  
}else{  
    x=x+1;  
}
```

```
switch(entrada){  
    case 1:  
        x=1;  
        break;  
    case 2:  
        x=2;  
        break;  
}
```

```
for(int j=1;j<10;j=j+1){  
}  
  
while(1){  
}
```


Operadores - conjunto de caracteres que permitem operar sobre variáveis ou constantes. As mais usadas em c++ são:

Aritméticos:

+ Adição
- Subtração
* Multiplicação
/ Divisão

Lógicos:

&& Conjunção (e)
|| Disjunção (ou)
== Igual a
!= Diferente de
! Negação
> Maior que
< Menor que

Atribuição:

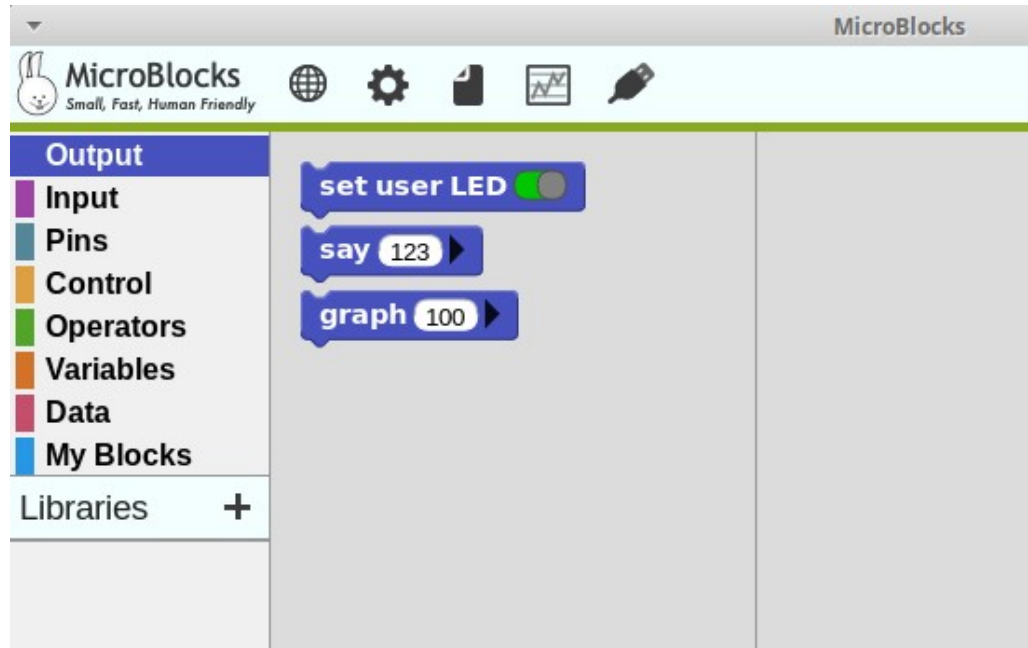
= atribui um valor a
uma variável

Exemplo - Acender e apagar um LED

```
#define LED 2
```

```
void setup() {  
  // Set pin mode  
  pinMode(LED,OUTPUT);  
}  
void loop() {  
  delay(500);  
  digitalWrite(LED,HIGH);  
  delay(500);  
  digitalWrite(LED,LOW);  
}
```

AzoresBot



<https://microblocks.fun/>