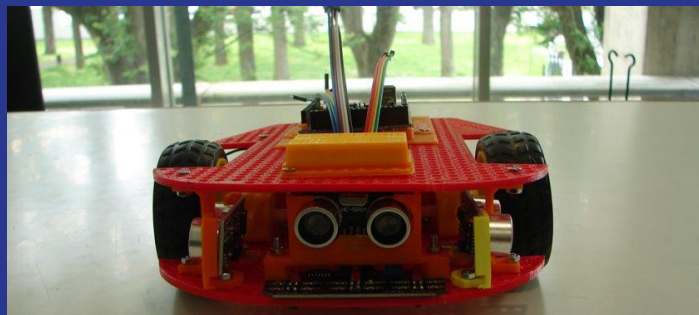
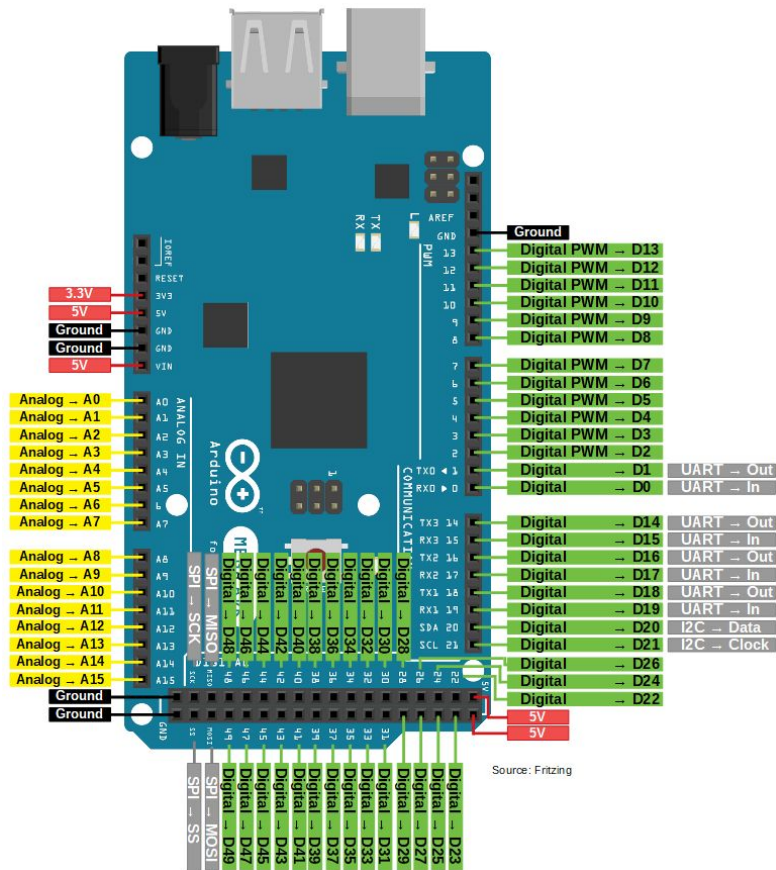


Programação e Robótica para um robô Arduino

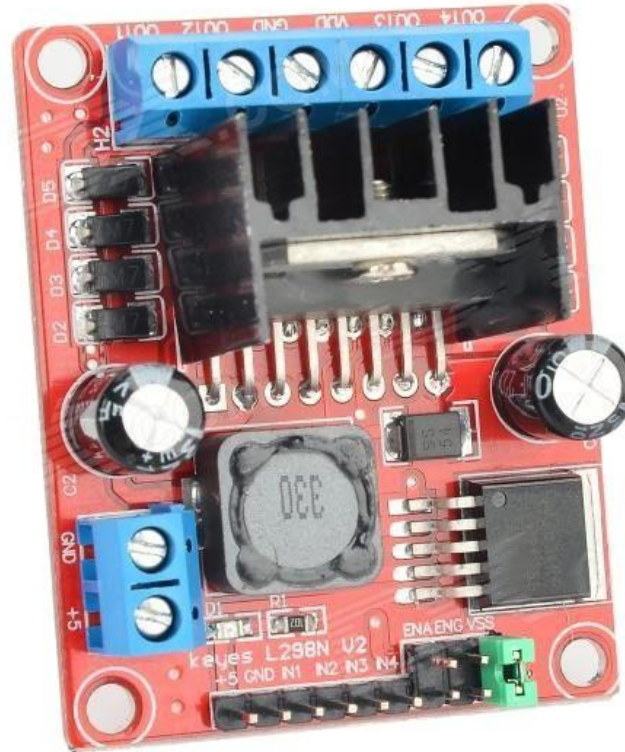
AZORESBOT



Hardware



Hardware



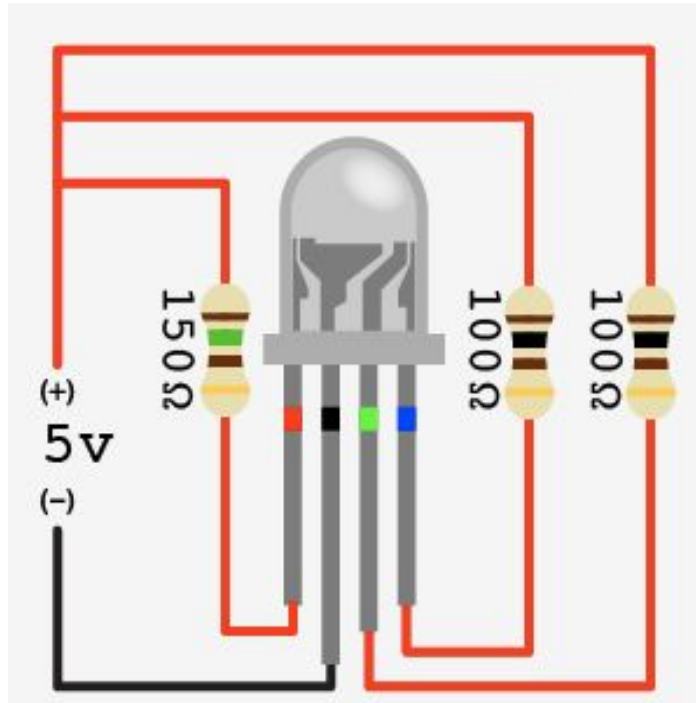
Hardware



Hardware



Hardware



Plataforma Arduino

- Arduino é uma plataforma electrónica de código aberto baseada em hardware e software fáceis de usar;
- As placas do Arduino são capazes de ler entradas - luz em um sensor, um dedo em um botão ou uma mensagem no Twitter - e transformá-lo em uma saída - ativando um motor, ligando um LED, publicando algo online;
- Podemos dizer à placa o que fazer enviando um conjunto de instruções ao microcontrolador na placa;
- Mais informação em <https://www.arduino.cc>

Plataforma Arduino

- O IDE oficial do Arduino é gratuito e pode ser descarregado no seguinte link :
<https://www.arduino.cc/en/Main/Software>

O Arduino IDE permite realizar as seguintes ações:

- Criar o código de um programa para o Arduino;
- Salvar o código do programa no computador;
- Compilar o código do programa;
- Carregar o código do programa para a placa Arduino;
- Comunicar com a placa Arduino;

Plataforma Arduino - Programação

- A linguagem de programação usada no 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;
- No caso das placas Arduino podemos usar o Arduino IDE que possui um compilador integrado.

Plataforma Arduino - Programação

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:  
  
}
```

Plataforma Arduino - Programação

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.

Plataforma Arduino - Programação

No menu do IDE clique em Ficheiro > Exemplos > Basics > Blinks.

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                     // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);                     // wait for a second  
}
```

Plataforma Arduino - Programação

Um programa para arduino pode ser dividido nas seguintes partes:

- Variáveis
- Funções
- Estruturas

Plataforma Arduino - Programação

Variáveis

- Servem para armazenar dados num programa;
- Os dados podem ser de diferentes tipos, como por exemplo:

int (número inteiro de 16 bits com sinal);
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);

Plataforma Arduino - Programação

Para usar uma variável necessitamos 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

Plataforma Arduino - Programação

Quando declaramos uma variável o seu valor é indeterminado.

Quando lhe atribuimos um valor estamos a inicializar a variável;

```
int x = 56;
```

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

```
boolean sensor = true
```


Plataforma Arduino - Programação

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
```

Plataforma Arduino - Programação

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;  
}
```

Plataforma Arduino - Programação

Funções

Para o arduino existe um conjunto de funções que controlam a placa e realizam computação - <https://www.arduino.cc/reference/pt/> - como por exemplo:

`pinMode()`

`analogRead()`

`digitalRead()`

`analogWrite()`

`digitalWrite()`

`delay()`

`pulseIn()`

`map()`

Plataforma Arduino - Programação

Instruções de controlo de execução:

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

```
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){  
}
```

Plataforma Arduino - Programação

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

Plataforma Arduino - Programação

Bibliotecas

Por forma a simplificar a construção de programas, usam-se bibliotecas – programas produzidos normalmente para interagir com sensores ou atuadores. Para as usarmos devemos incluí-las no nosso programa usando o comando `#include`, e de seguida devemos instanciar os objetos necessários, por exemplo:

```
#include <Motor.h>  
Motor motor_direito(10,9,8);  
Motor motor_esquerdo(5,6,7);
```

Plataforma Arduino - Programação

Depois de instanciados os objetos podemos usá-los no nosso sketch em conjunto com as funções da classe em causa:

```
void loop(){  
  motor_direito.avancar(255);  
  motor_esquerdo.avancar(255);  
}
```

Neste exemplo a classe Motor tem 3 funções avançar(velocidade), recuar(velocidade) e parar().