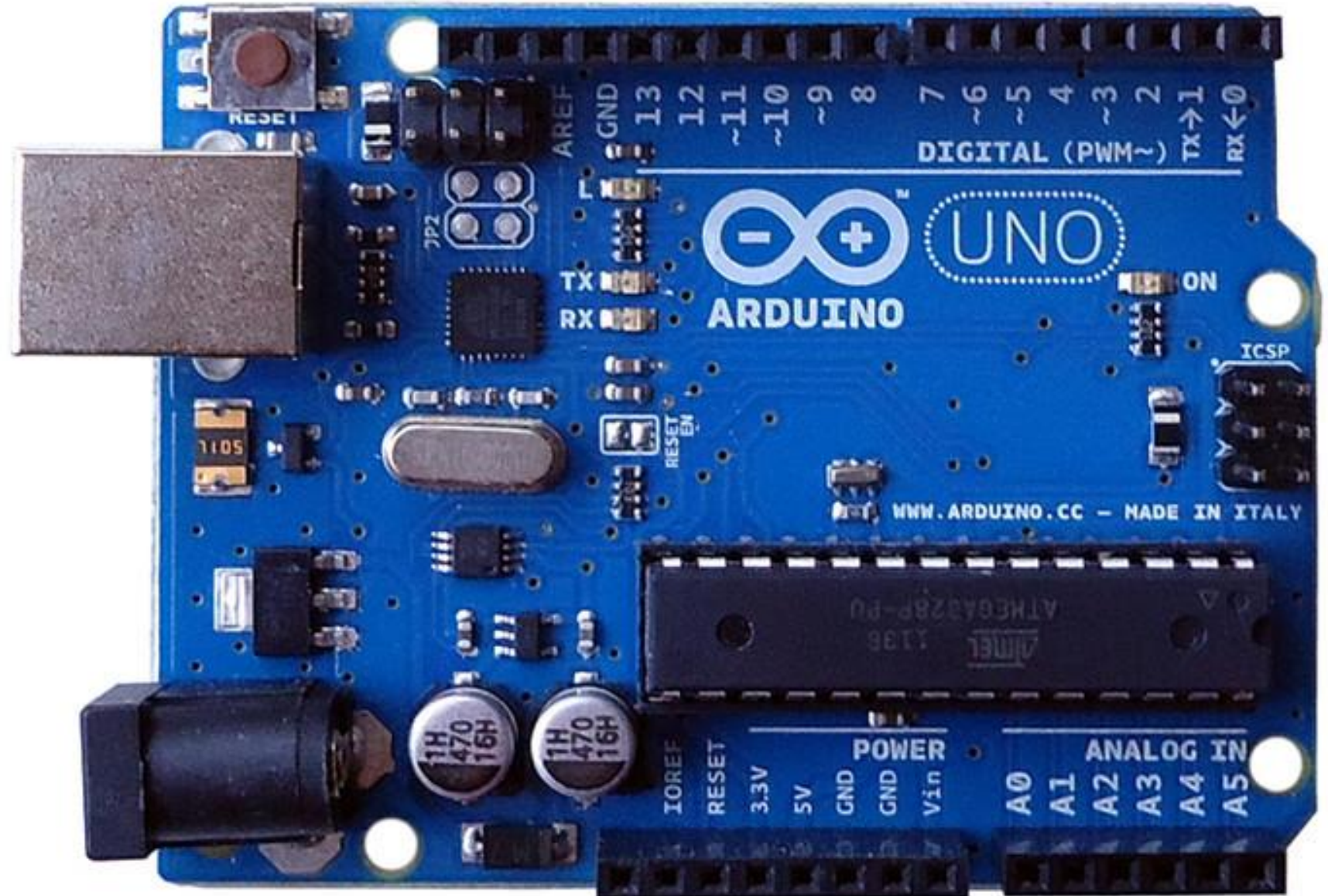


Microprocessadores e Microcontroladores

Introdução à prática com a placa Arduino Uno

José Tarcísio Franco de Camargo

O que é um Arduino?



O que é o Arduino?

- O Arduino é uma plataforma eletrônica de código aberto, sendo ideal para qualquer pessoa que deseja realizar a prototipagem de projetos eletrônicos.
- As placas Arduino permitem a conexão de circuitos eletrônicos aos seus terminais, o que possibilita a leitura de entradas/informações:
 - Luz/temperatura/posição/orientação proveniente de um sensor, etc.
- Também permitem transformar informações em algum tipo de saída:
 - Acionamento de um LED, de um motor, etc.

O que é um Arduino?

- As placas Arduino podem ser conectadas a um computador por meio de uma porta USB.
- Isso possibilita a sua utilização como placa de interface, bem como permite controlar dispositivos por meio de um computador.

Vantagens de uma placa Arduino

- Possui ambiente multiplataforma, ou seja, pode ser executado nos principais sistemas operacionais.
- Conta com uma interface de programação própria.
- Pode ser programado utilizando um cabo USB.
- Possui hardware e software de fonte aberta.
- Foi desenvolvido em um ambiente educacional, sendo ideal para iniciantes em automação, robótica, etc.

Versões de uma placa Arduino



ARDUINO UNO



ARDUINO LEONARDO



ARDUINO 101



ARDUINO ESPLORA



ARDUINO MICRO



ARDUINO NANO



ARDUINO DUE



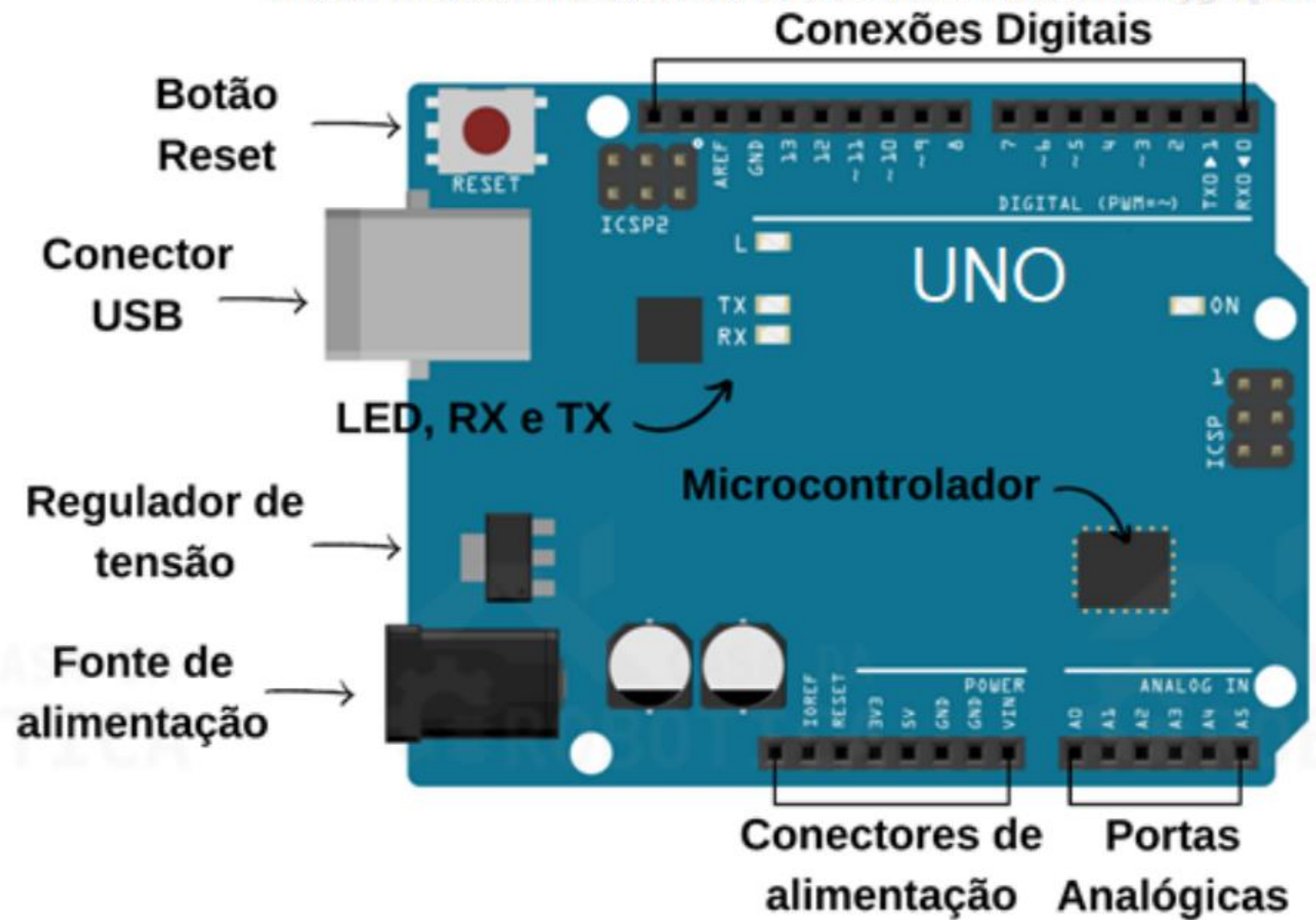
ARDUINO MEGA 2560



ARDUINO ZERO

Explorando uma placa Arduino Uno R3

Placa microcontroladora UNO SMD R3 ATMEGA238.



Explorando uma placa Arduino Uno R3

- **Fonte de alimentação:**

- O circuito interno da placa Uno deve ser alimentado com uma tensão contínua de 5V.
- Você pode alimentá-lo conectando-o a uma porta USB do computador, que fornecerá a alimentação e também a comunicação de dados.
- Também pode alimentar o Arduino por meio de uma fonte de alimentação externa, que forneça uma tensão contínua entre 7 V e 12 V no plug P4.
- A mesma fonte de alimentação externa pode ser conectada por meio da utilização do pino **Vin** do Arduino.

- **Regulador de tensão:**

- Na placa Uno, o regulador de tensão tem como finalidade transformar qualquer tensão de entrada (entre 7 V e 12 V), que esteja sendo fornecida pelo conector de alimentação, em uma tensão contínua de 5V.

Explorando uma placa Arduino Uno R3

- **Conectores de alimentação elétrica:** Os conectores de alimentação elétrica fornecem energia para dispositivos externos e são constituídos pelos pinos:
 - **Reset** que possui a mesma função do botão **Reset**;
 - **3,3 V** e **5 V** que fornecem tensão de 3,3 e 5 V, respectivamente;
 - **GND** que fornece potencial de terra aos dispositivos externos;
 - **Vin** que fornece ao dispositivo externo a mesma tensão que está sendo recebida pelo pino de alimentação externa.

Explorando uma placa Arduino Uno R3

- **Entradas analógicas:**

- A placa UNO possui 6 portas analógicas que estão indicados como **Analog In**, de **A0** a **A5**.
- Esses pinos são dedicados a receber valores de grandezas analógicas, por exemplo, a tensão de um sensor.
- As grandezas analógicas variam continuamente no tempo dentro de uma faixa de valores.

Explorando uma placa Arduino Uno R3

- **Conexões digitais:**

- A placa UNO possui 14 portas digitais que estão indicados como **Digital**, de 0 a 13.
- Estas portas podem ser utilizadas para receber ou enviar dados de grandezas digitais.
- Ao contrário das grandezas analógicas, as grandezas digitais não variam continuamente no tempo, mas sim em saltos entre valores definidos (0 ou 1, ligado ou desligado, sim ou não, 0 V ou 5 V).
- Estes pinos digitais operam em 5V e corrente máxima de 40 mA.
- Alguns desses pinos possuem funções especiais:
 - Pinos 3, 5, 6, 9, 10 e 11 podem ser usados como saídas PWM;
 - Pinos 0 e 1 (RX e TX) podem ser utilizados para comunicação serial;
 - Pinos 2 e 3 podem ser configurados para gerar uma interrupção externa.

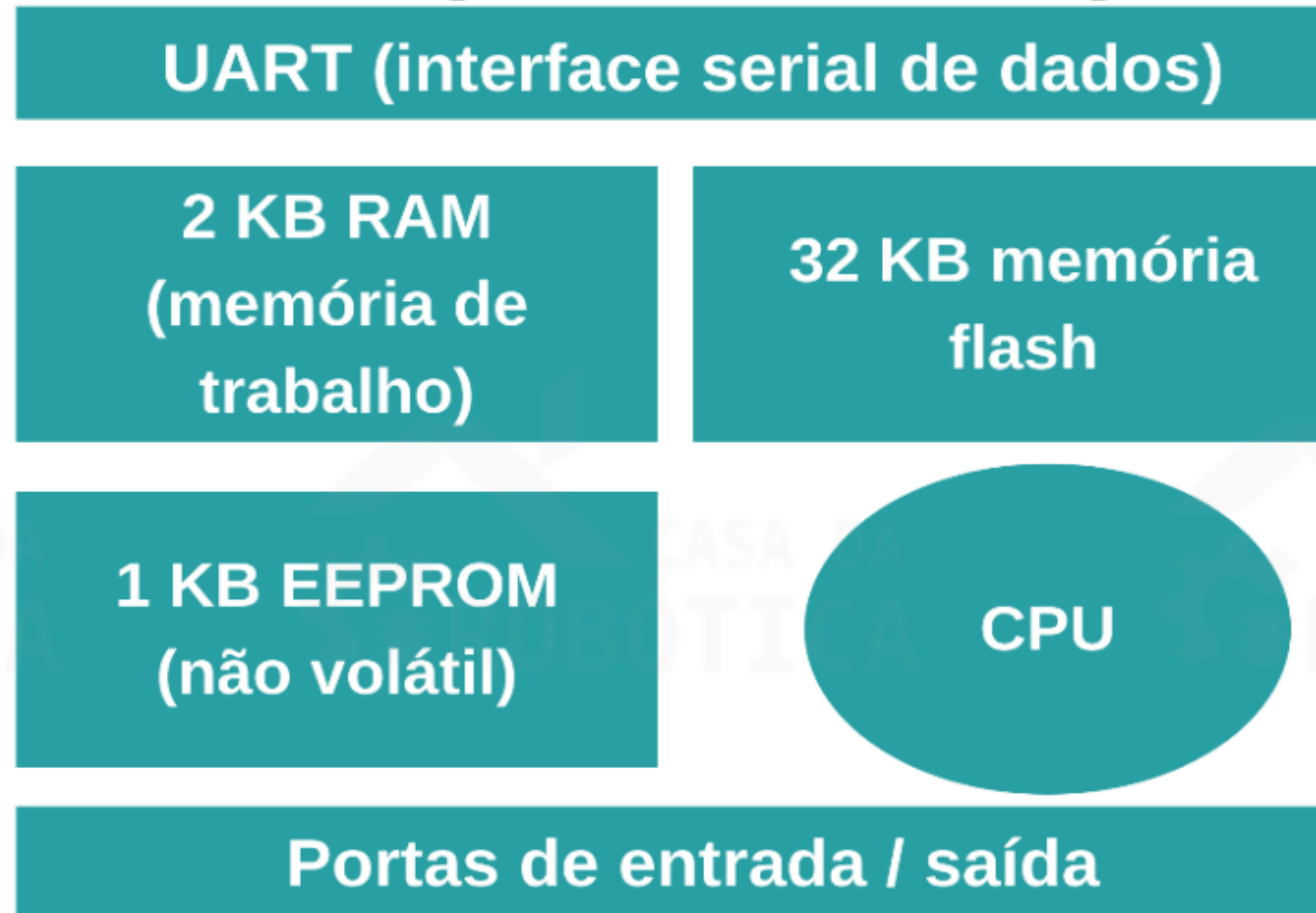
Explorando uma placa Arduino Uno R3

- **Microcontrolador:**

- O microcontrolador utilizado na placa UNO é o ATmega328, um chip de 28 pinos que se encontra no centro da placa e é considerado o “cérebro” dela.
- Esse único chip é um pequeno computador, contendo memória, processador e toda eletrônica necessária aos pinos de entrada e saída.
- É no microcontrolador que tudo acontece, é nele que fica gravado o código desenvolvido para execução.
- O microcontrolador permite que a placa Uno funcione de forma autônoma, em outras palavras, uma vez transferido o código não existe mais a necessidade de comunicação com o computador.
- Um fato que deve ser lembrado é que ao gravar um código, o anterior é descartado, ficando apenas o último código gravado.

Características do microcontrolador ATmega328

Diagrama de blocos do ATmega328.



Explorando uma placa Arduino Uno R3

- **Botão Reset:**

- O botão Reset tem como única função reinicializar a placa Arduino.

- **Outros componentes:** Além dos componentes citados, a placa UNO também conta com:

- um oscilador a cristal, capaz de realizar 16 milhões de ciclos ou oscilações por segundo;
- conector serial de programação, (outro meio de programar a placa UNO);
- chip de interface USB, que converte os níveis de sinal usados pelo padrão USB em níveis que podem ser usados pela placa Uno.

Introdução à programação com Arduino

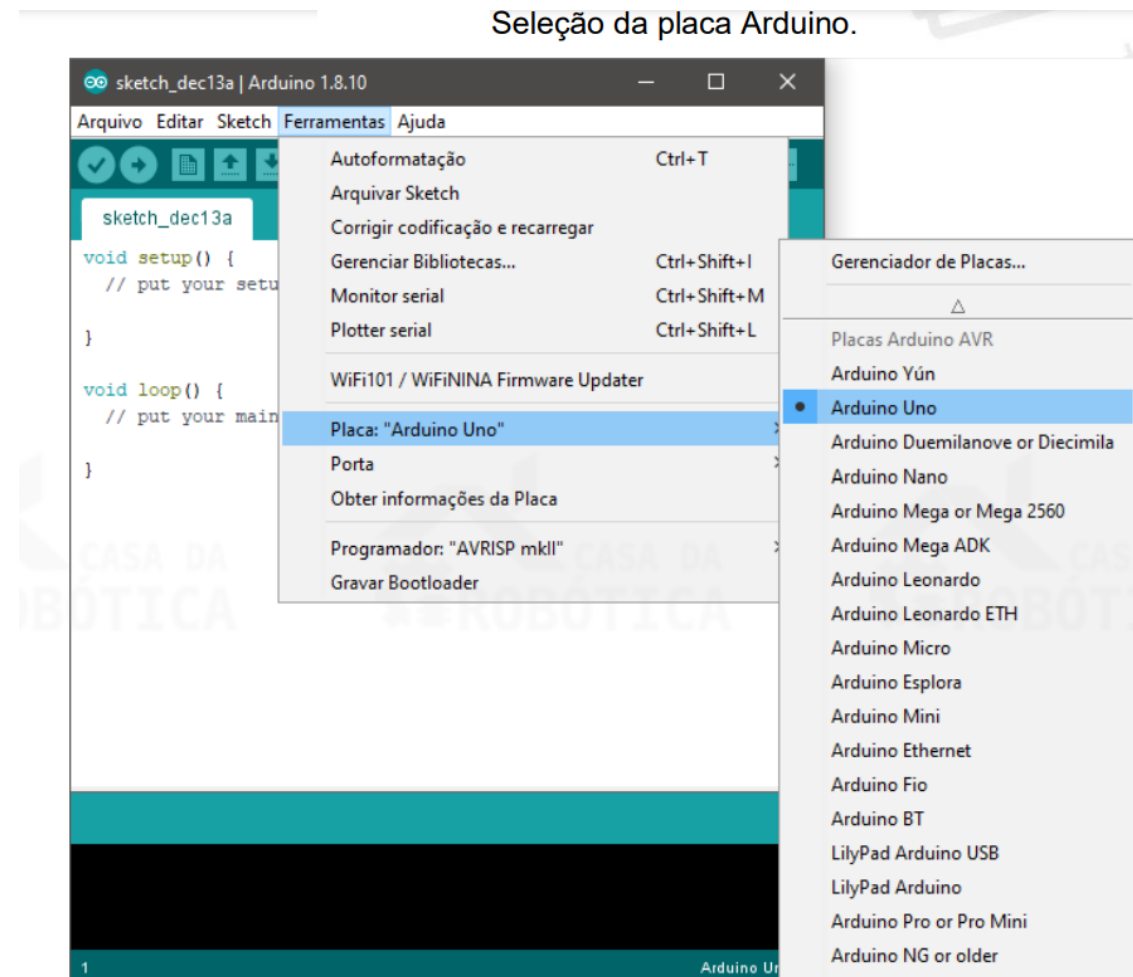
- Ao abrir o Arduino IDE você verá uma tela semelhante à figura abaixo.

Visual do Arduino IDE.

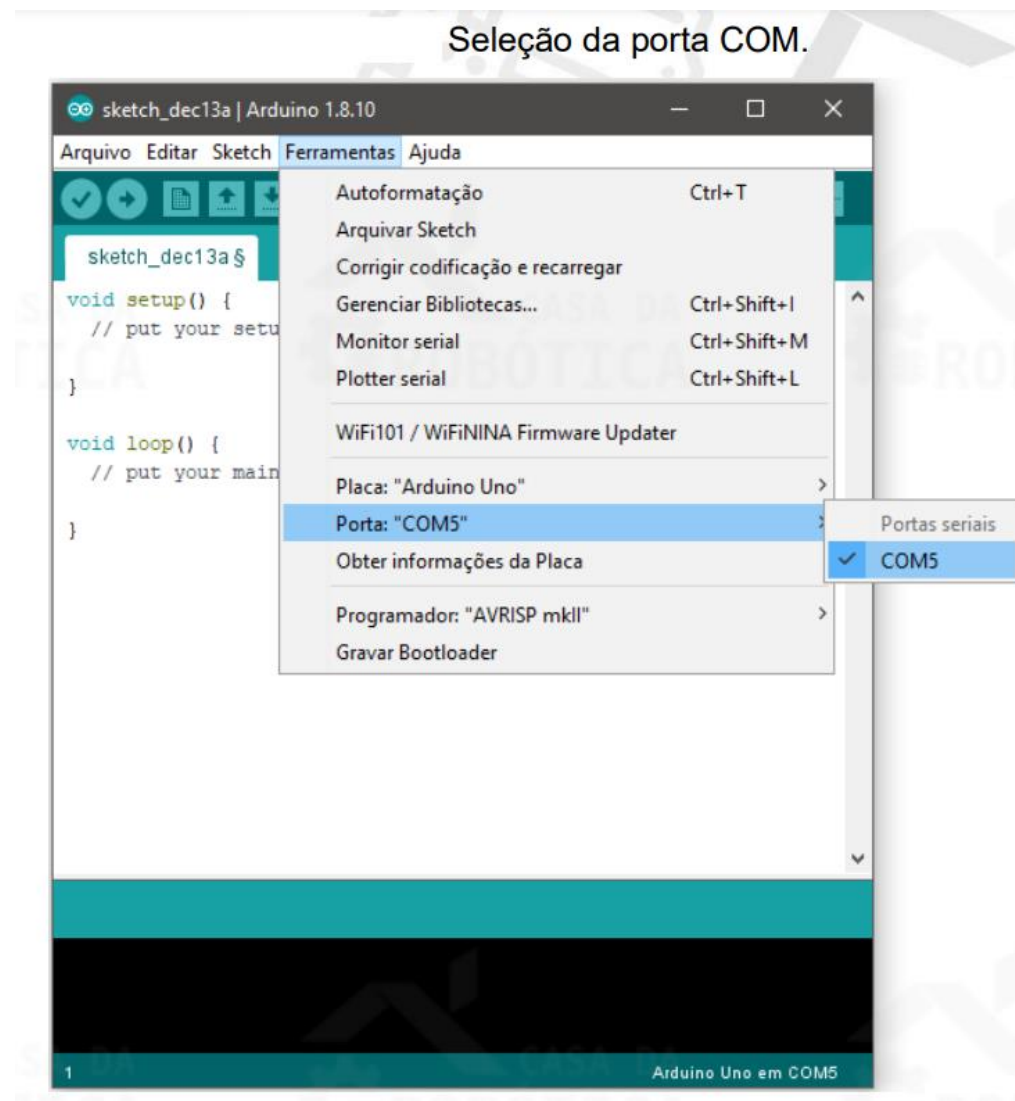


Introdução à programação com Arduino

- Após conectar a placa Arduino ao computador, é preciso selecionar o tipo de placa e a porta de comunicação no Arduino IDE.



Introdução à programação com Arduino



Estrutura básica de um “sketch”

- Todos os sketches Arduino devem ter a estrutura composta pelas funções **setup()** e **loop()**, conforme ilustra o exemplo a seguir.

```
1  void setup()  
2  {  
3      Comando 1;  
4      Comando 2;  
5      ...  
6  }  
7  void loop()  
8  {  
9      Comando 3;  
10     Comando 4;  
11     ...  
12 }
```

A função `setup()` é executada apenas uma vez na inicialização do programa, e é nela que você deverá descrever as configurações e instruções gerais para preparar o programa antes que o loop principal seja executado. Em outras palavras, a função `setup()` é responsável pelas configurações iniciais da placa microcontroladora, tais como definições de pinos de entrada e saída, inicialização da comunicação serial, entre outras.

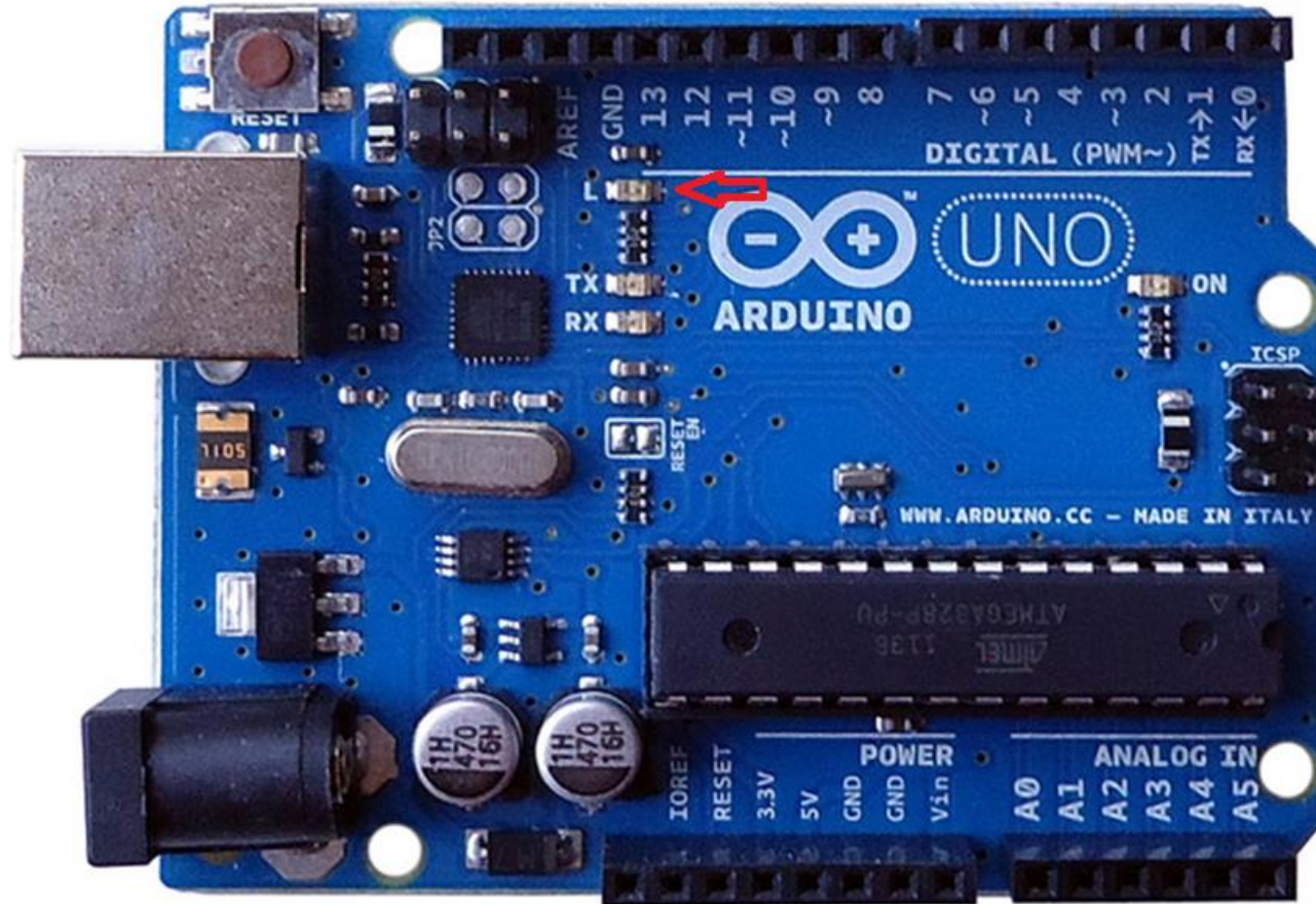
A função `loop()` é a função principal do programa e é executada continuamente enquanto a placa microcontroladora estiver ligada. É nesta função que todos os comandos e operações deverão ser escritos.

Primeiro programa – Piscar LED da placa Uno

- O exemplo mais básico e clássico para iniciar a programação do Arduino é o “Blink” (ou Pisca Led), que consiste em acionar um LED por meio de um sinal digital.

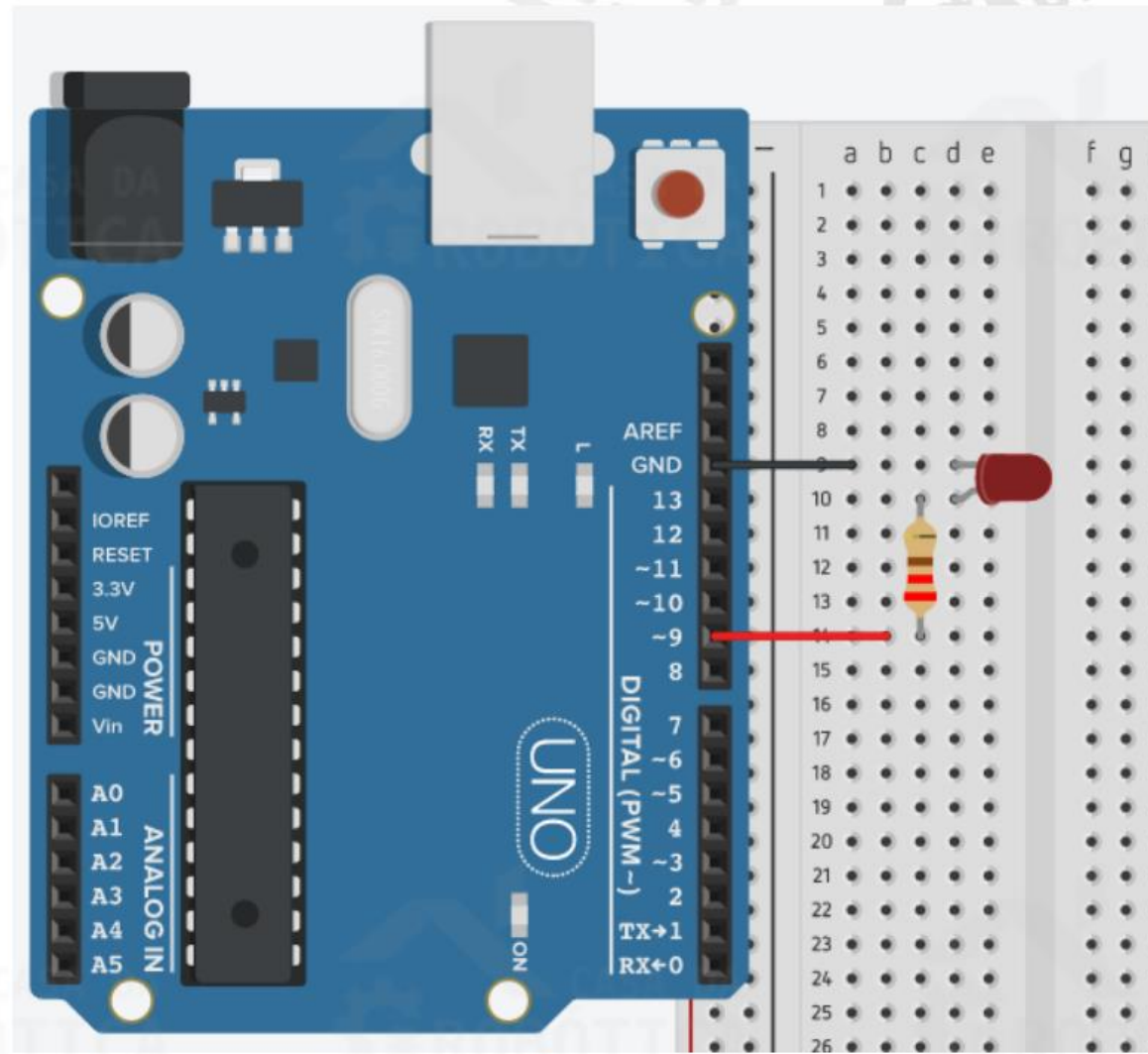
```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage  
  level)  
  delay(1000); // wait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

Primeiro programa – Piscar LED da placa Uno



Segundo programa – piscar um LED externo

Circuito para o projeto Piscar o LED.



Segundo programa – piscar um LED externo

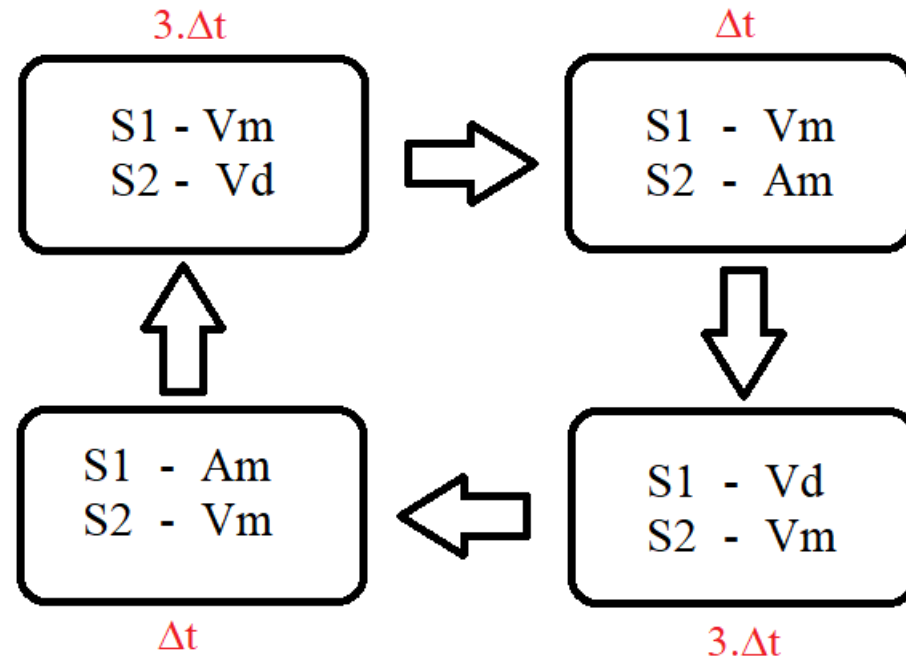
```
// Projeto - Piscar LED
int ledPin = 9; //Atribui o valor 9 a variável inteira ledPin, que irá
representar o pino digital 9

void setup() {
    pinMode(ledPin, OUTPUT); //Define ledPin (pino 9) como saída
}

void loop() {
    digitalWrite(ledPin, HIGH); //Coloca ledPin em nível alto (5V)
    delay(1000); //Espera 1000 milissegundos (1 segundo)
    digitalWrite(ledPin, LOW); //Coloca ledPin em nível baixo (0V)
    delay(1000); //Espera 1000 milissegundos (1 segundo)
}
```

Sua tarefa!

- Monte um circuito que possua dois “semáforos” de três cores cada (vermelho, amarelo e verde) e escreva um programa através da IDE do Arduino para que o semáforo funcione de maneira coerente.



Duração do Verde: $3.\Delta t$
Duração do Amarelo: Δt
Duração do Vermelho: $3.\Delta t + \Delta t$