

INTRODUÇÃO AO ARDUINO

- ◉ Capacitação em robótica ativa
- ◉ Pedro Graças Alves Júnior - Coordenador de conteúdo (TRON)

O QUE É O ARDUINO?



- Plataforma de prototipagem eletrônica *open source* (*hardware e software* livres).
- Destinado a qualquer pessoa interessada em criar objetos ou ambientes interativos.
- Ele pode perceber o ambiente ao seu redor recebendo em suas entradas uma variedade de sensores e pode modificar sua vizinhança controlando luzes, motores, displays, e etc.

O QUE SIGNIFICA ARDUINO?

- Derivado de um termo de origem germânica, *harduwin* ou *hardwin*, composto de hardu "forte, resistente" e wini "amigo";
- "Forte" porque são raros os relatos de quem tenha queimado um Arduino;
- "Amigo" porque é simples de mexer e não demanda muito conhecimento.

ORIGEM DO ARDUINO?

- Criado na cidade de Ivrea, Itália em 2005;

Criadores:

- Massimo Banzi,
- David Cuartielles,
- Tom Igoe,
- Gianluca Martino e
- David Mellis.

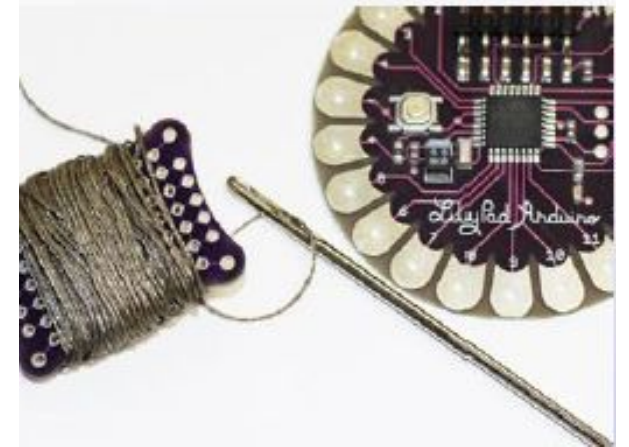


PLACAS DE ARDUINO?

- Arduino Mega



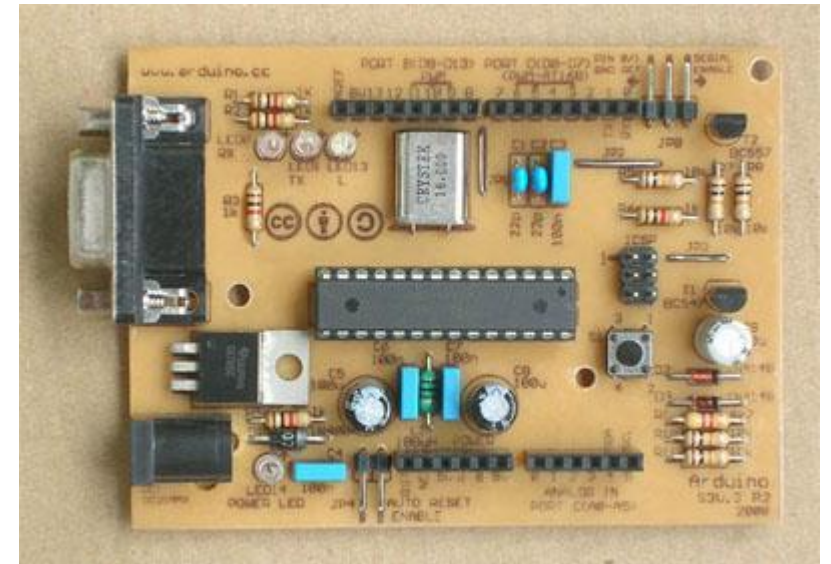
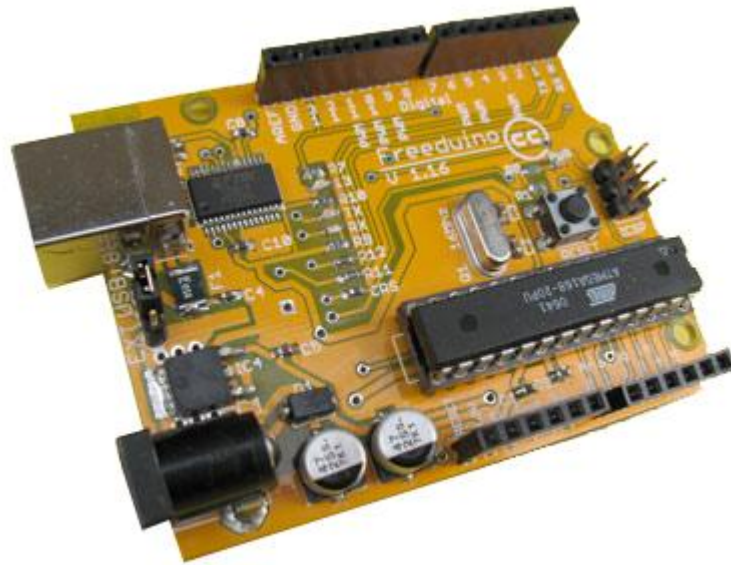
Arduino Lilypad



Arduino Nano

◎ CLONES DO ARDUINO

■ Freeduino, Severino, etc.



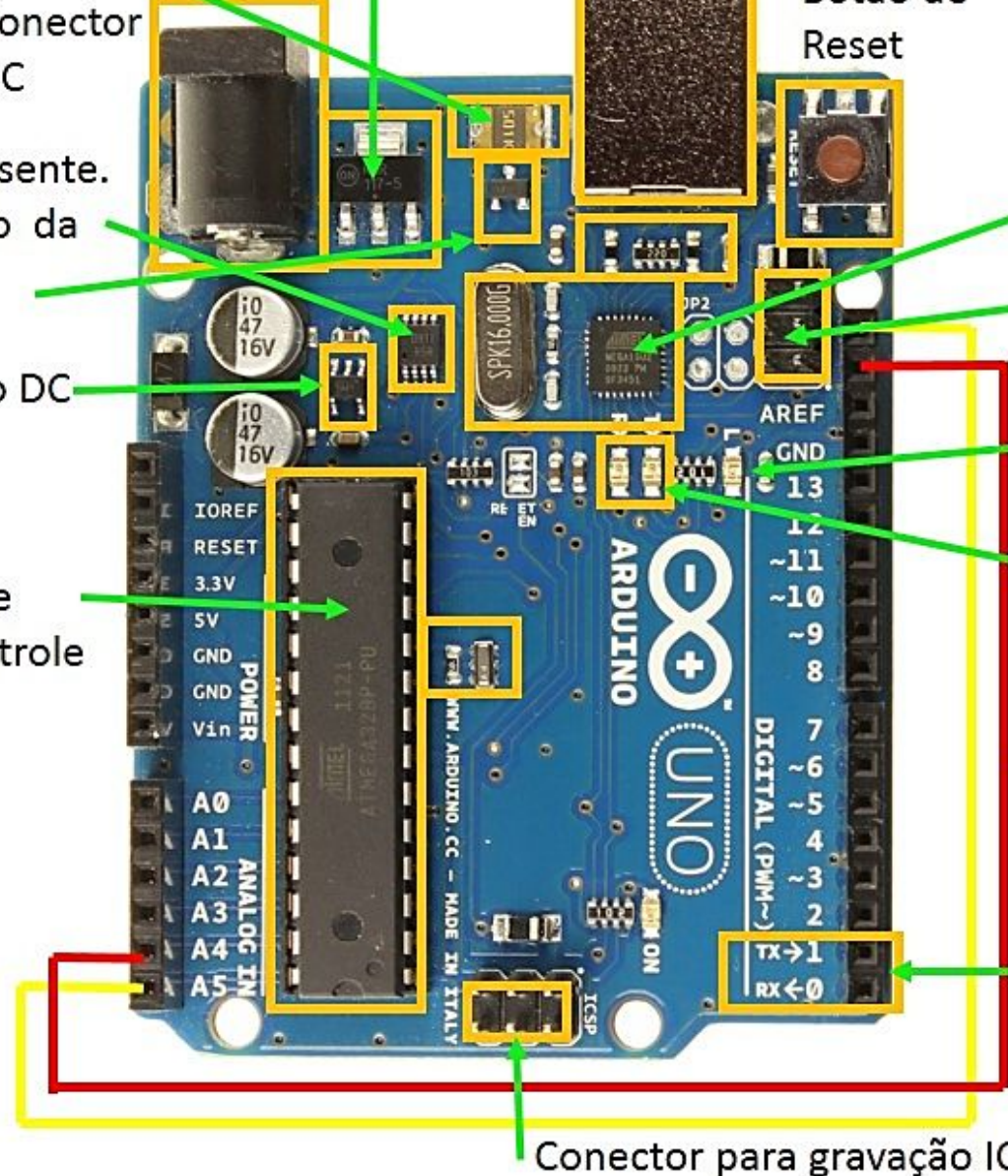
caso de sobrecorrente.
(acima de 500 mA)

Conector
DC

Compara se a tensão DC está presente.
Se não estiver, deixa que a tensão da
USB Alimente o circuito.

Regula a tensão DC
para 3,3 V.

Conjunto microcontrolador e
cristal, responsável pelo controle
e leitura de todos os pinos
da placa .



Reset

Conjunto microcontrolador e
cristal que faz a interface USB com
o computador

Conector para gravação ICSP,
do ATMEGA16U2

Led conectado ao pino 13
do arduino

Leds de status da comunicação serial
Entre placa e computador

Os sinais em amarelo e vermelho
Indicam dois pinos que estão em
curto

Caso utilize esses sinais no projeto,
tome cuidado pois estão conectados
ao outro microcontrolador para
gravação

Conector para gravação ICSP do ATMEGA328

**Fusível para
Proteção da USB**

FORNITORE
7 a 12V

Regulador 3,3 V

Saídas PWM

Pinos de alimentação

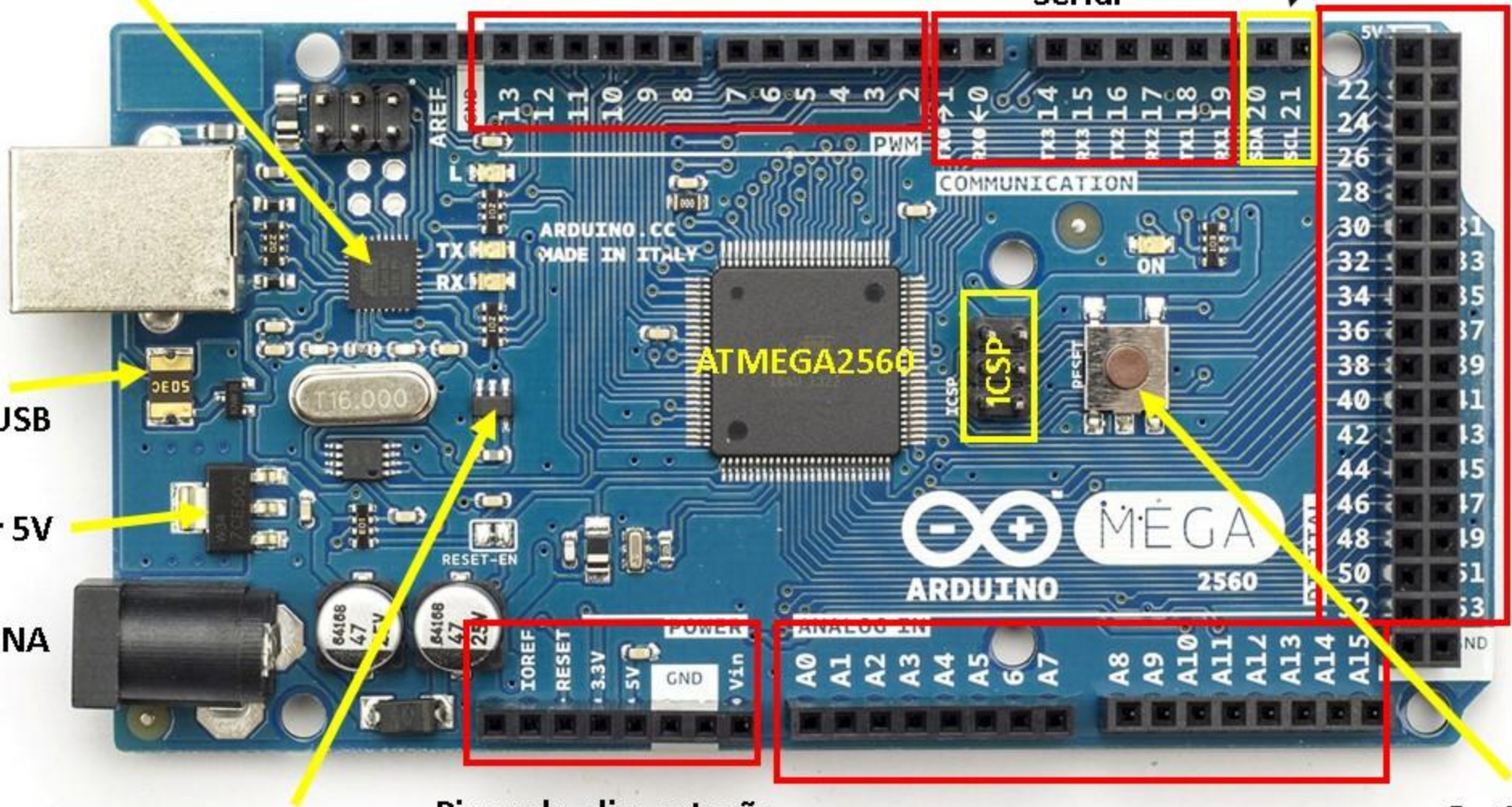
Comunicações Serial

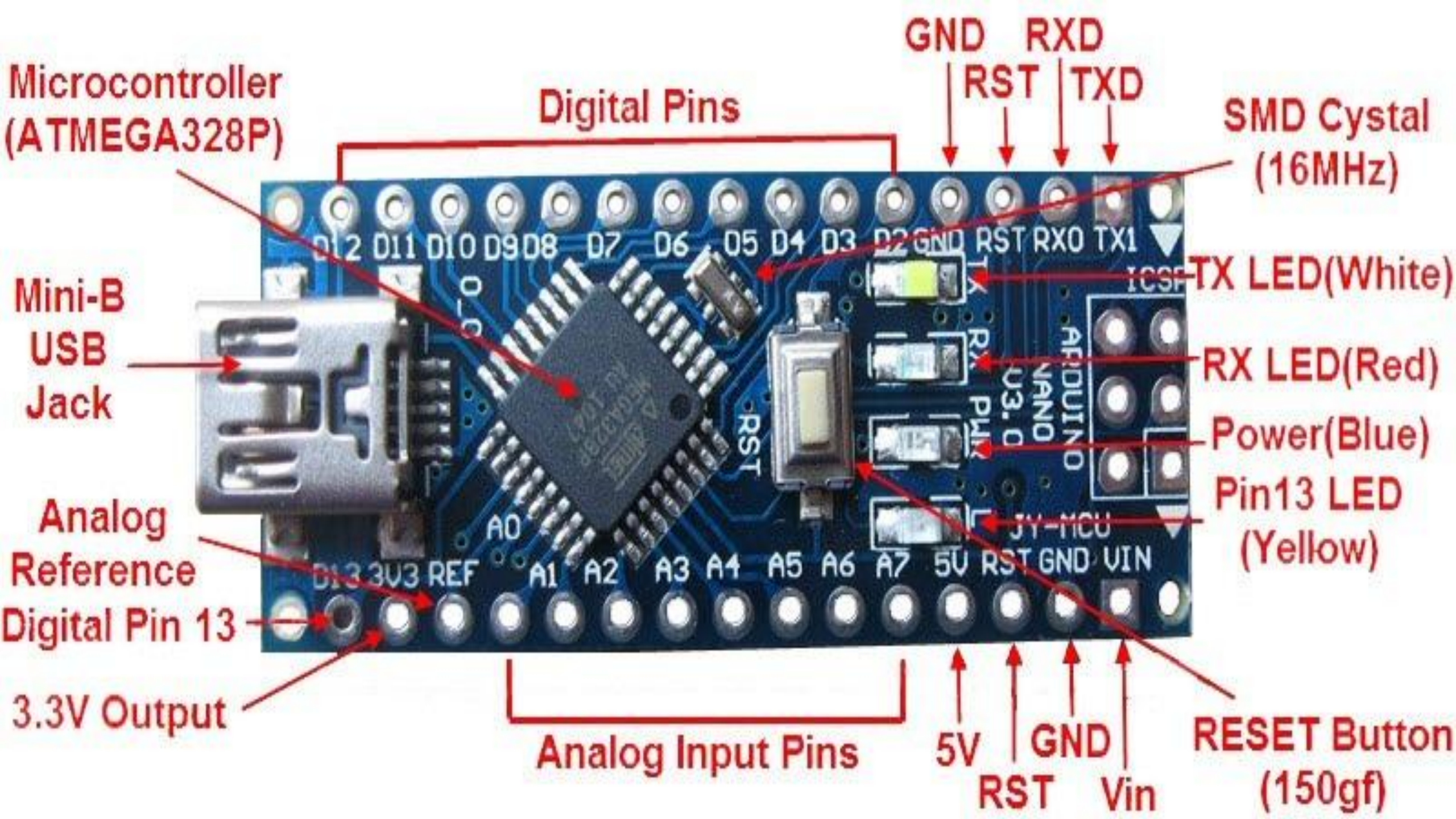
Entradas analógicas

TWI (I2C)

Entradas/Saídas Digitais de uso geral

Botão de RESET



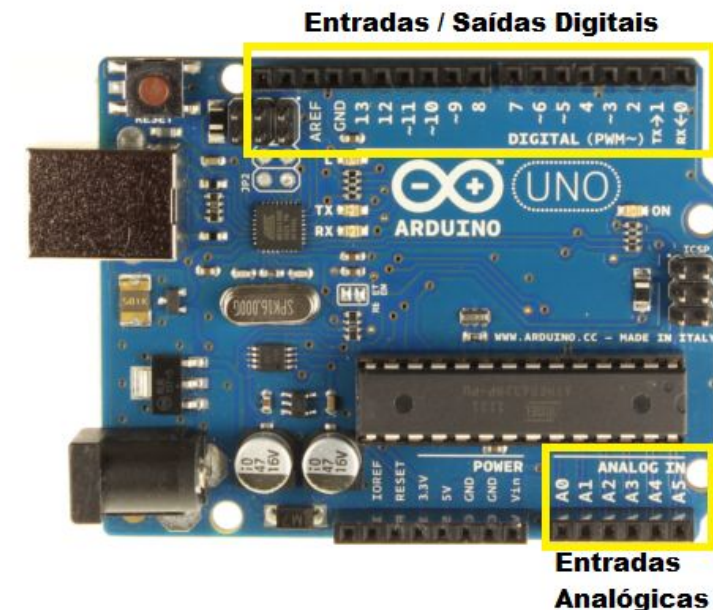


PORTAS DIGITAIS E ANALÓGICAS

O Arduino possui tanto portas digitais como portas analógicas.

As portas servem para comunicação entre o Arduino e dispositivos externos, por exemplo: ler um botão, acender um led ou uma lâmpada.

O Arduino UNO, possui 14 portas digitais e 6 portas analógicas (que também podem ser utilizadas como portas digitais).

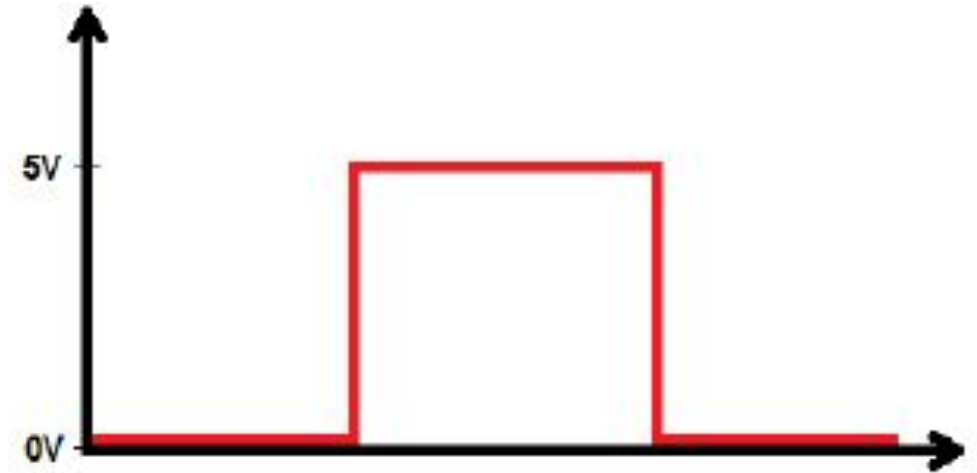


HARDWARE

BÁSICO

PORTAS DIGITAIS

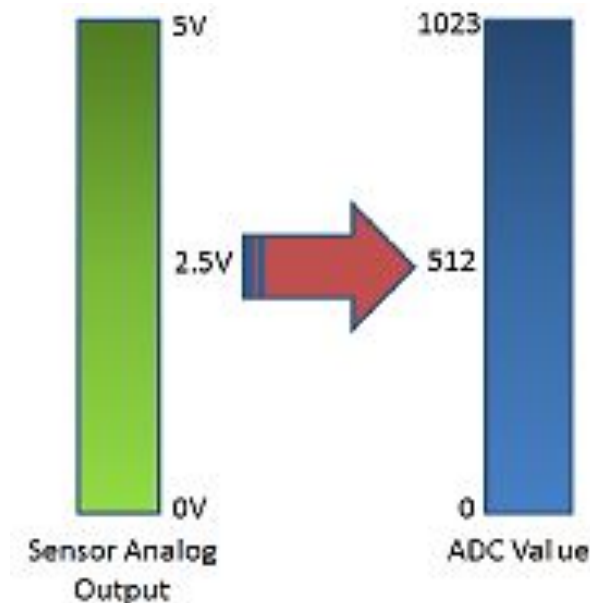
- *Desligado* ou *Ligado*
- 0 (zero) ou 5V
- *LOW* ou *HIGH*.



HARDWARE

PORTAS ANALÓGICAS BÁSICO

- Sinais de sensores com valores de 0 (zero) a 5V;
- Conversor de 10 bits com valores de 0 (zero) a 1.023.

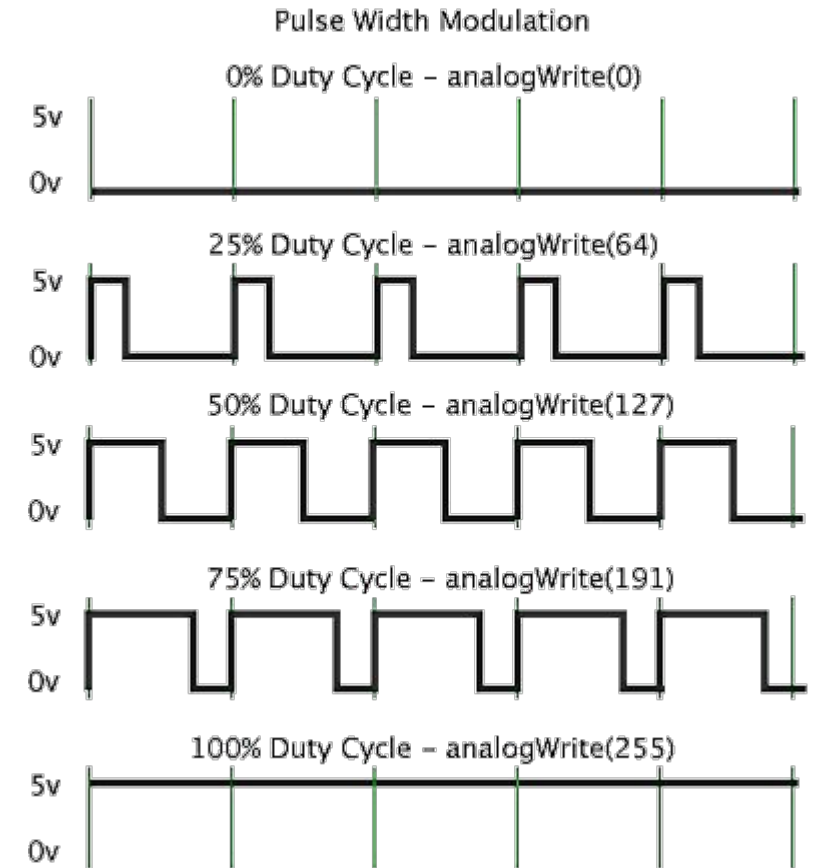


SAÍDAS

PWM

O que é PWM?

PWM, do inglês *Pulse Width Modulation* (*Modulação por Largura de Pulso*), é uma técnica utilizada por sistemas digitais para variação do valor médio de uma forma de onda periódica. A técnica consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. Esse tempo é chamado de *duty cycle*, ou seja, o ciclo ativo da forma de onda. No gráfico ao lado são exibidas algumas modulações PWM.

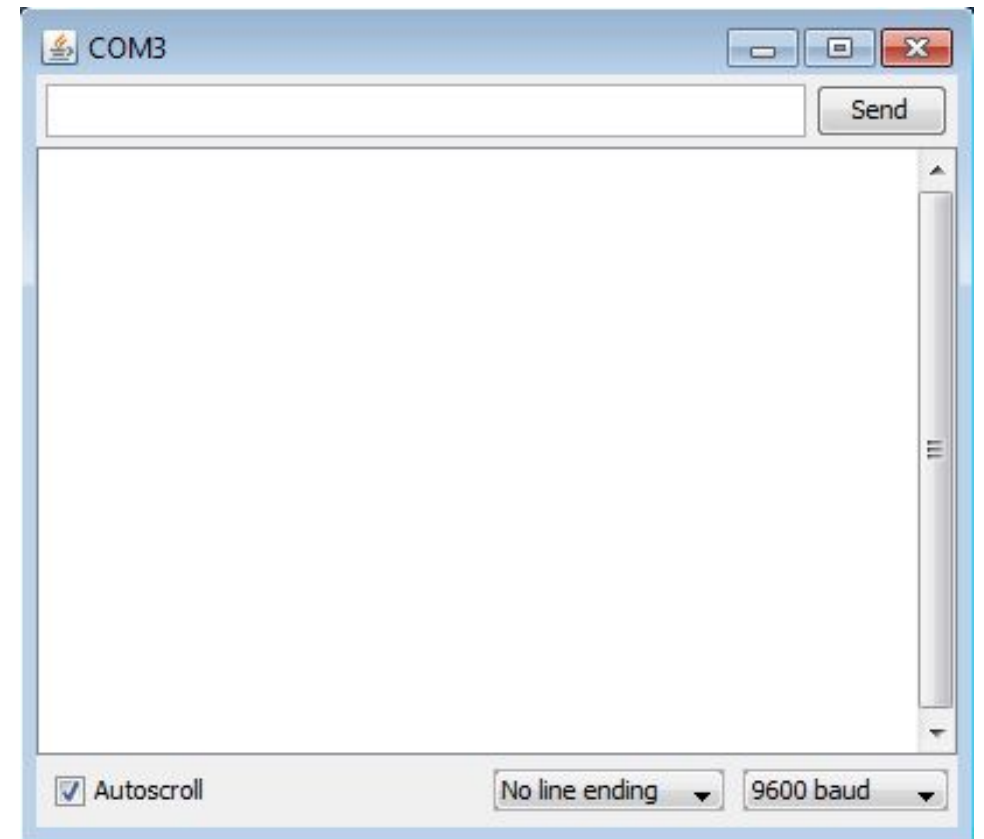


HARDWARE

BÁSICO

- Comunicação Serial

- RX (pino 0);
- TX (pino 1);

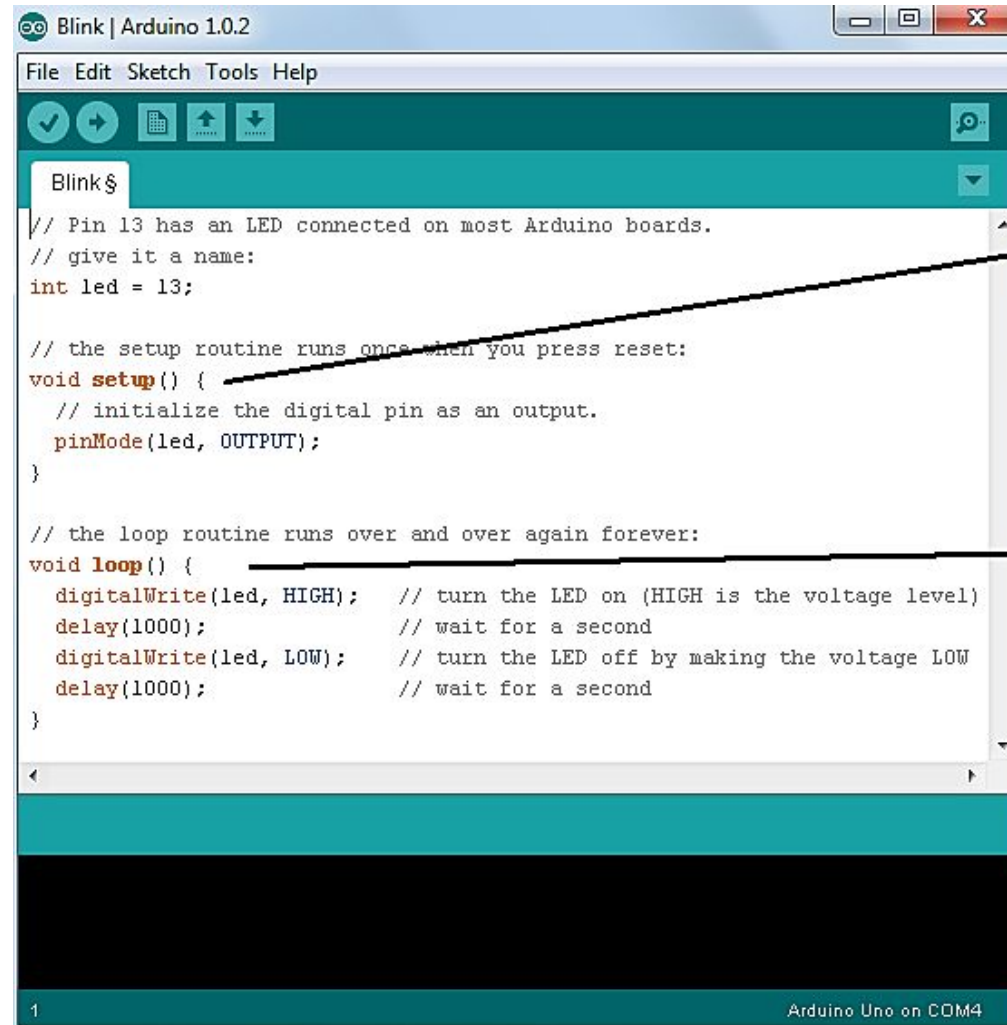


SOFTWARE

E

- O Arduino é programado por meio de uma linguagem própria muito semelhante à Linguagem C;
- Interface de desenvolvimento próprio (IDE – Interface de Desenvolvimento Integrada);
- Pode ser utilizada em vários sistemas operacionais.

Ambiente de Desenvolvimento ARDUINO



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is as follows:

```
Blink $  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

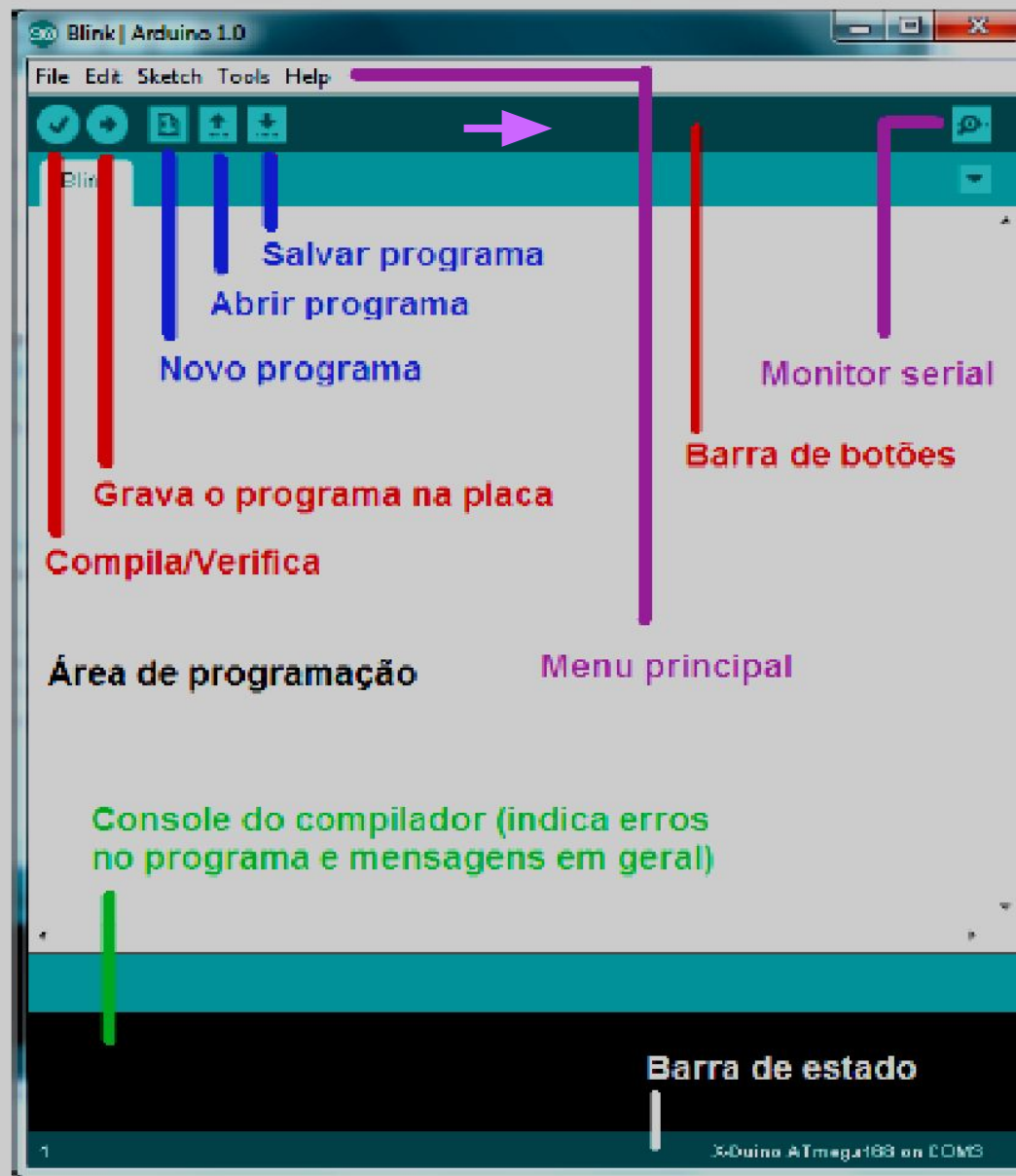
The IDE window title is 'Blink | Arduino 1.0.2'. The status bar at the bottom indicates '1' and 'Arduino Uno on COM4'.

SETUP

- * Configurações iniciais
- * O código é executado uma única vez

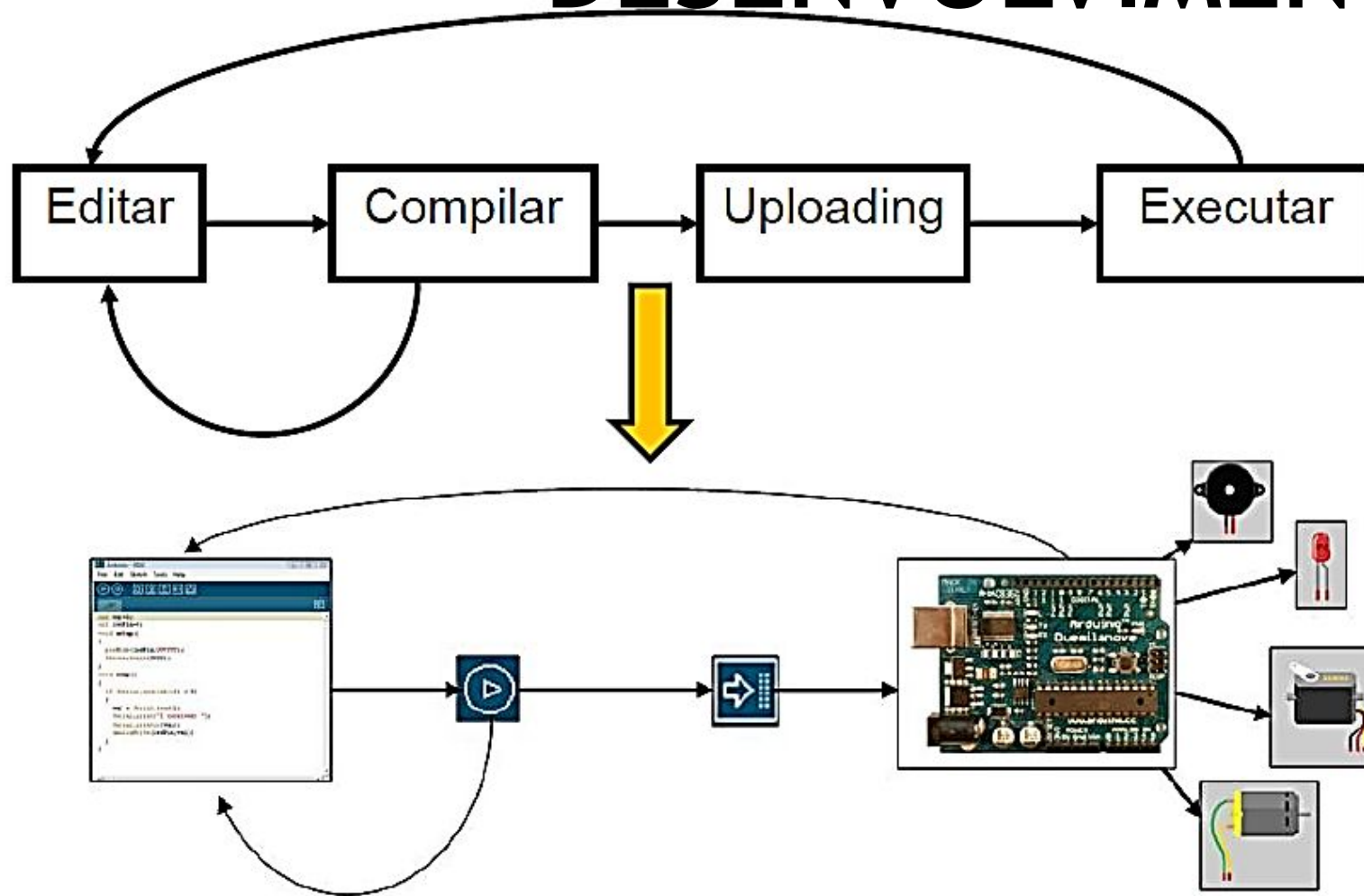
LOOP

- * Equivale ao Main do C
- * while infinito



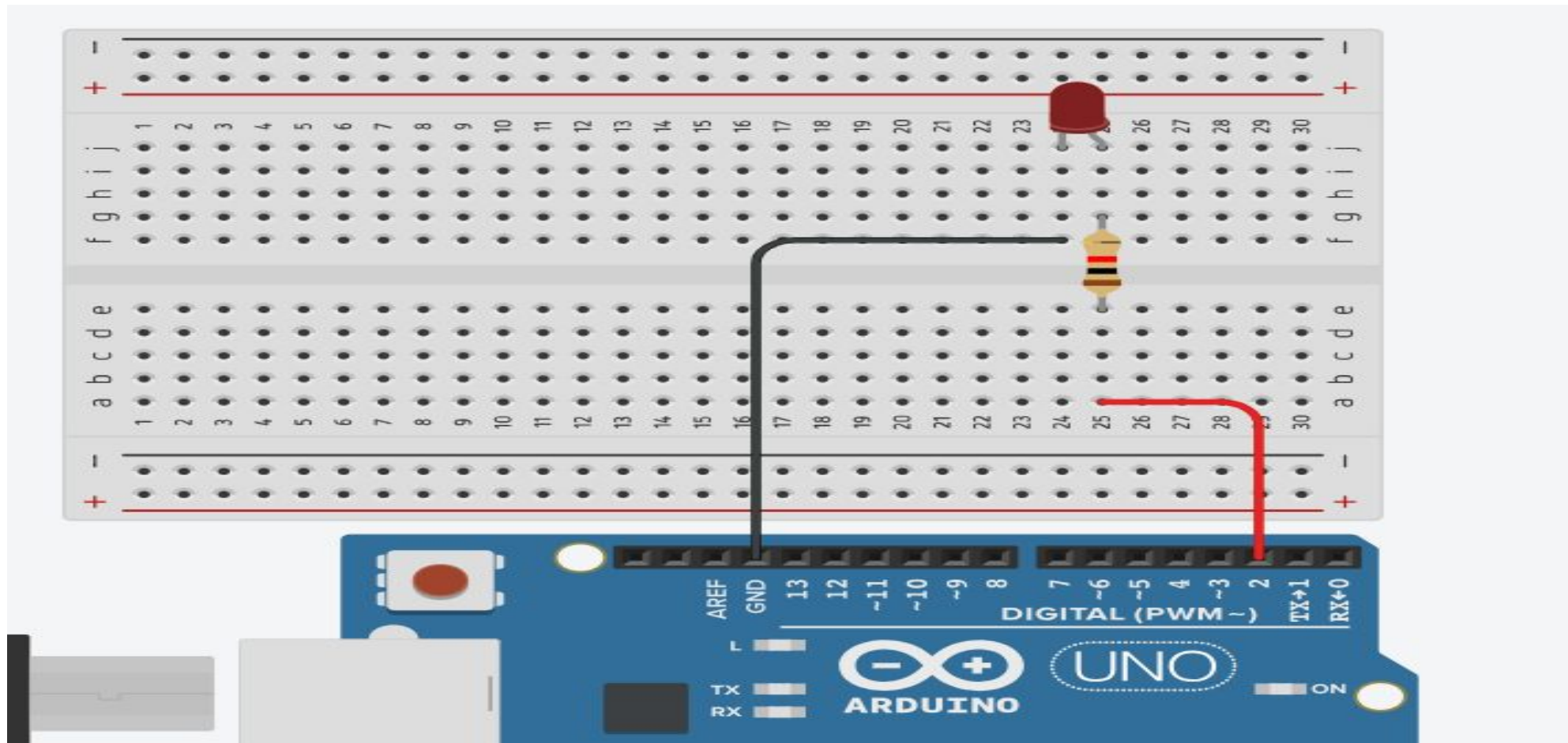
O monitor serial é utilizado para comunicação entre o Arduino e o computador (PC).

CICLO DE DESENVOLVIMENTO

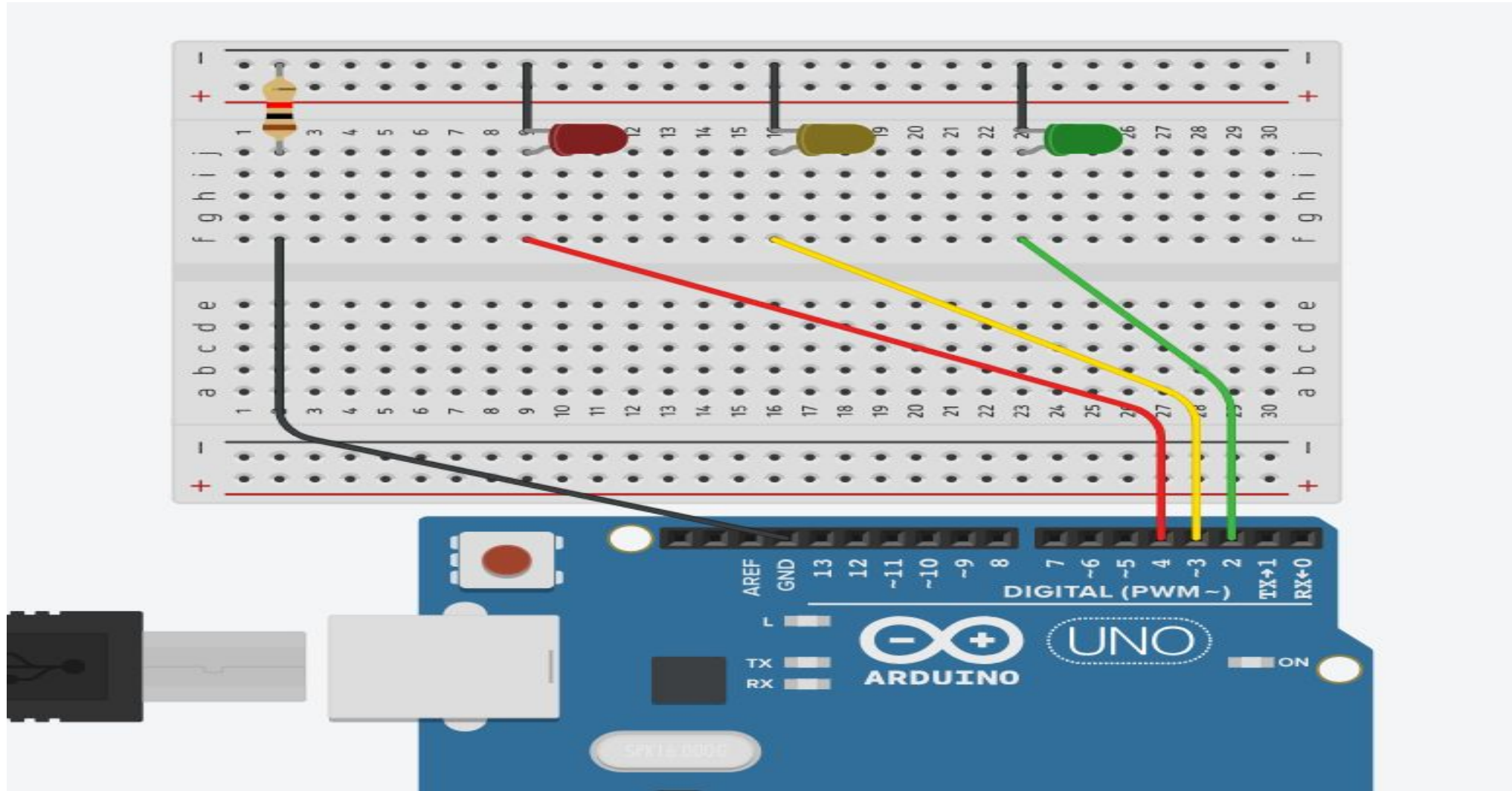


Programando com Arduino

Acendendo um LED:



Programando com Arduino



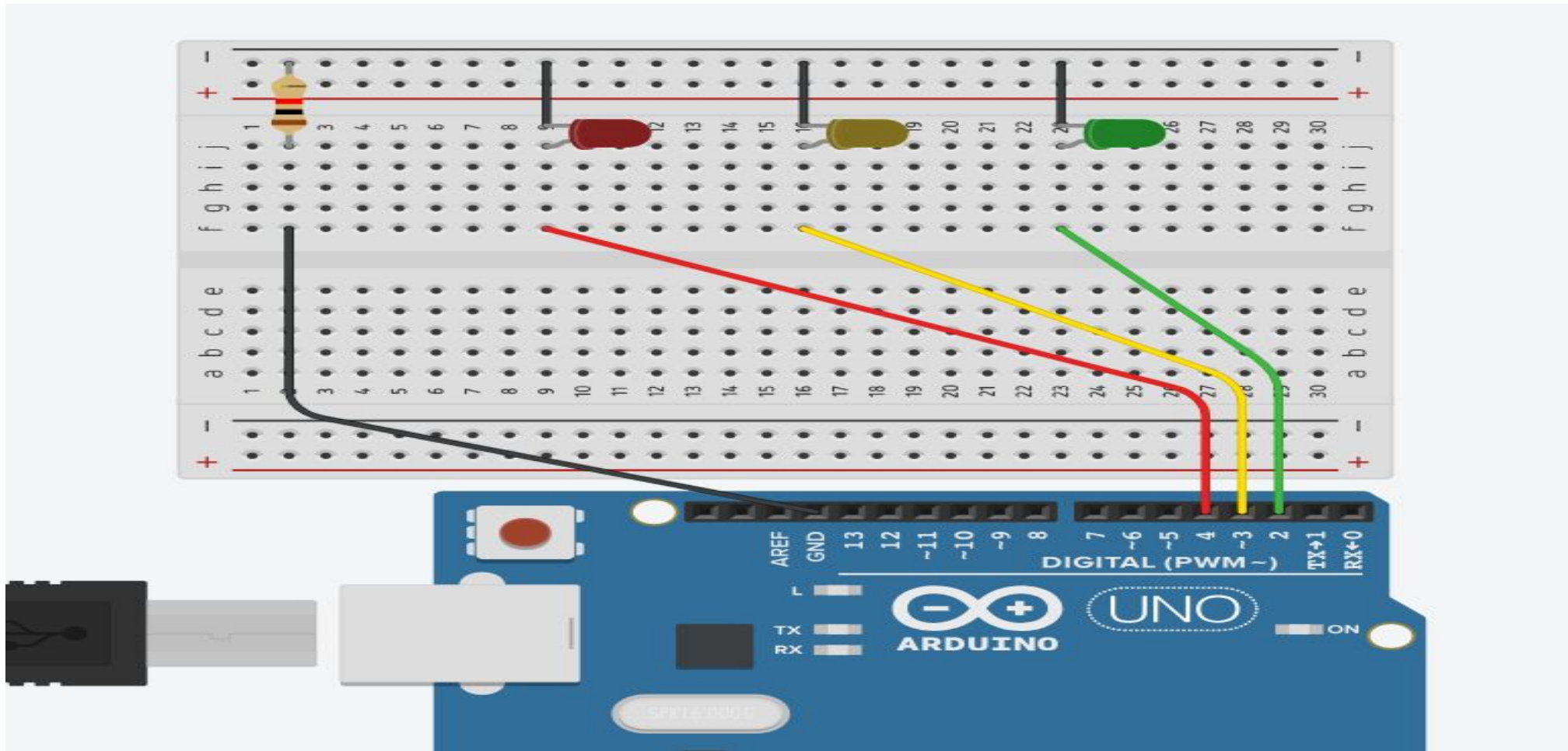
Programando com Arduino

De acordo com a imagem anterior resolva a lógica com LEDs:

- 1 - LED vermelho pisca 5X com delay de 500;
- 2 - LED amarelo pisca 4X com delay de 500;
- 3 - LED verde pisca 3X com delay de 500.

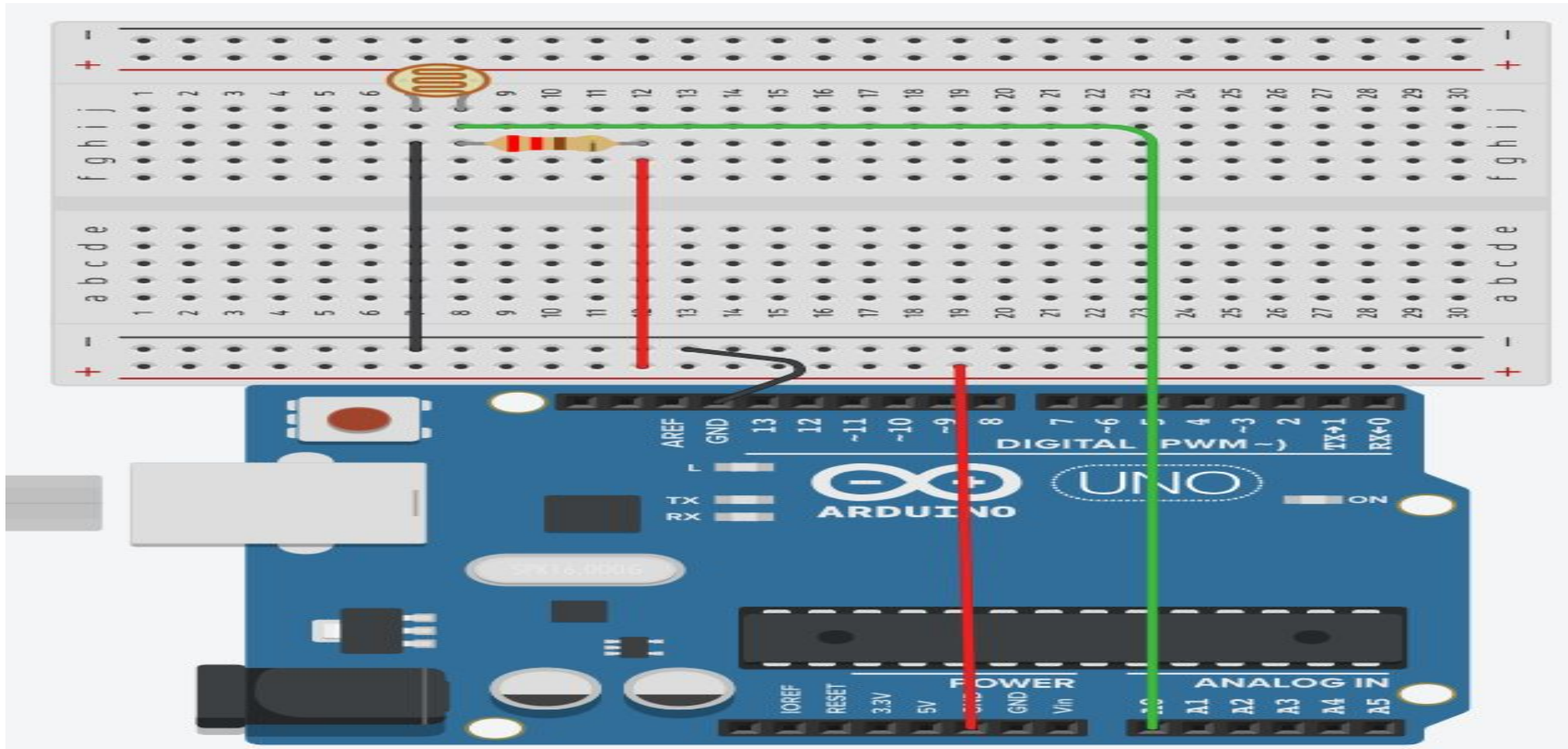
Programando com Arduino

Construa a lógica de um semáforo:



Programando com Arduino

Conhecendo sensores Analógicos e seu código:



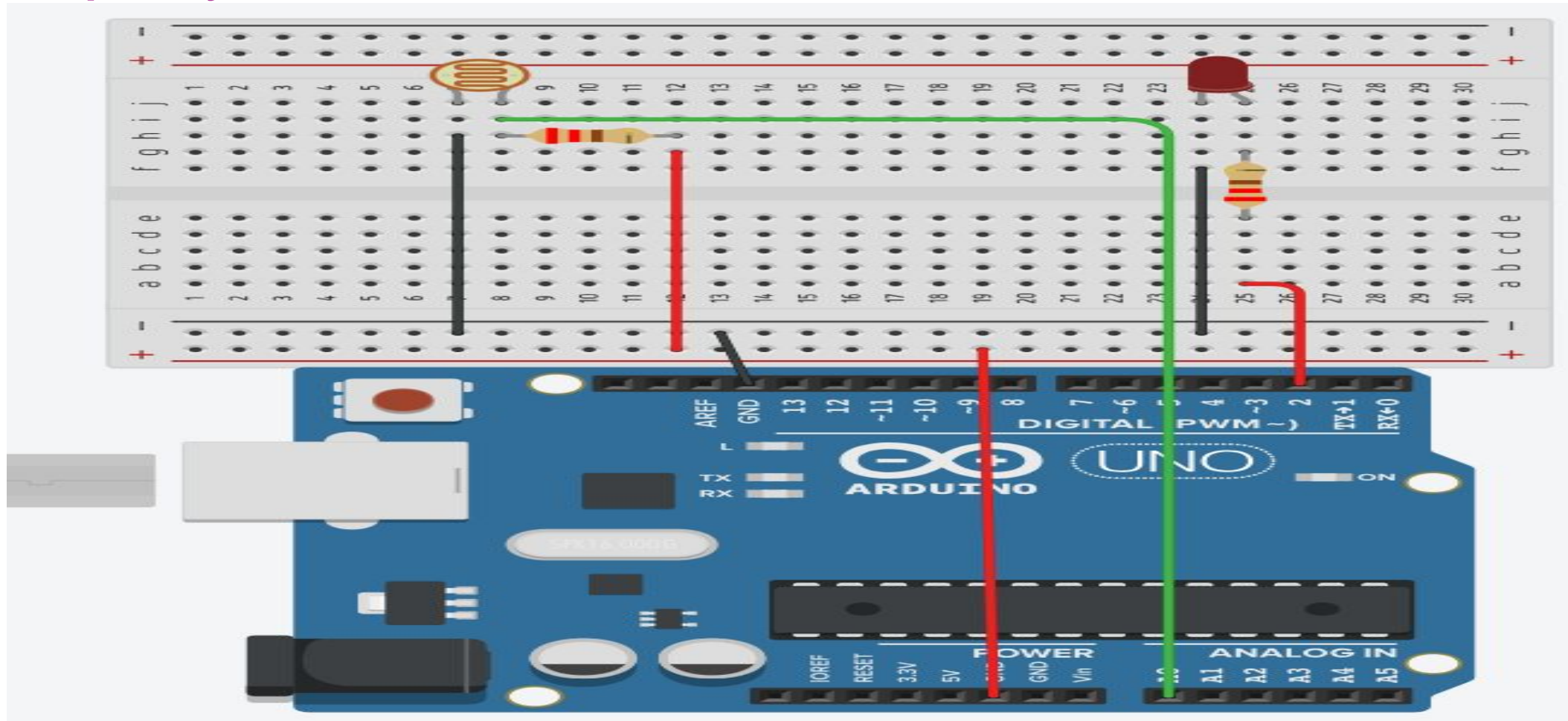
Programando com Arduino

Identificando taxas no monitor serial:



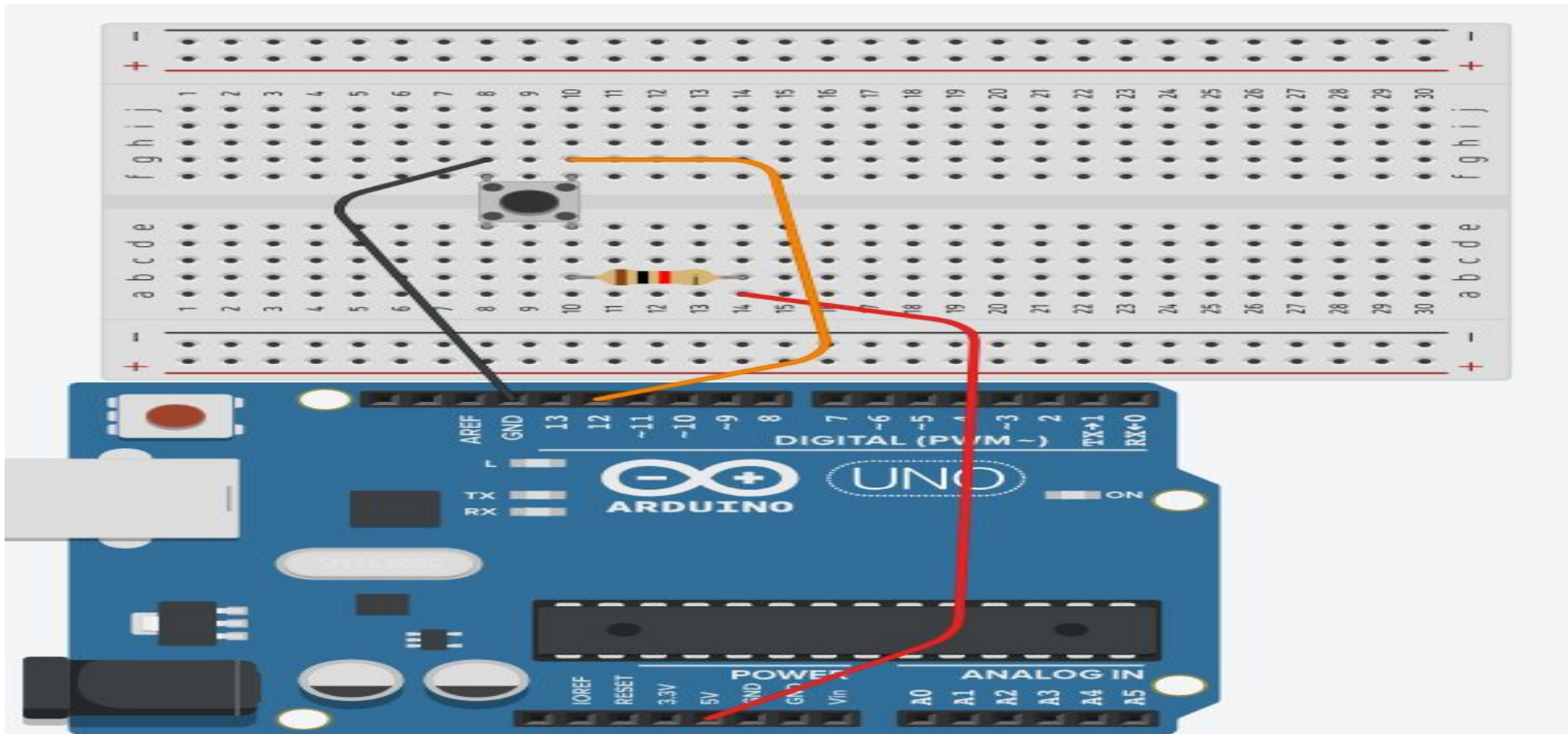
Programando com Arduino

Aplicação:



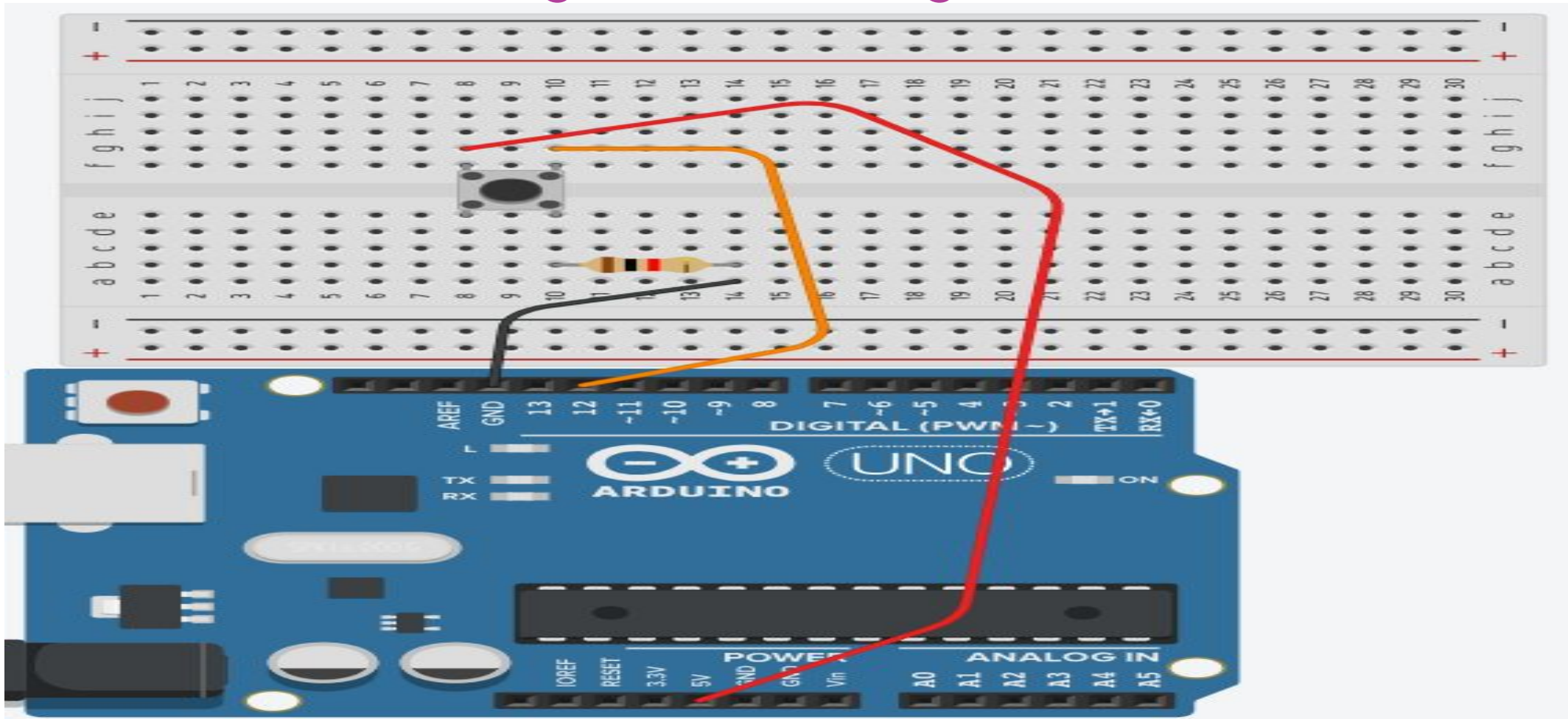
Programando com Arduino

Simulando sensores digitais e seu código - PULLDOWN:



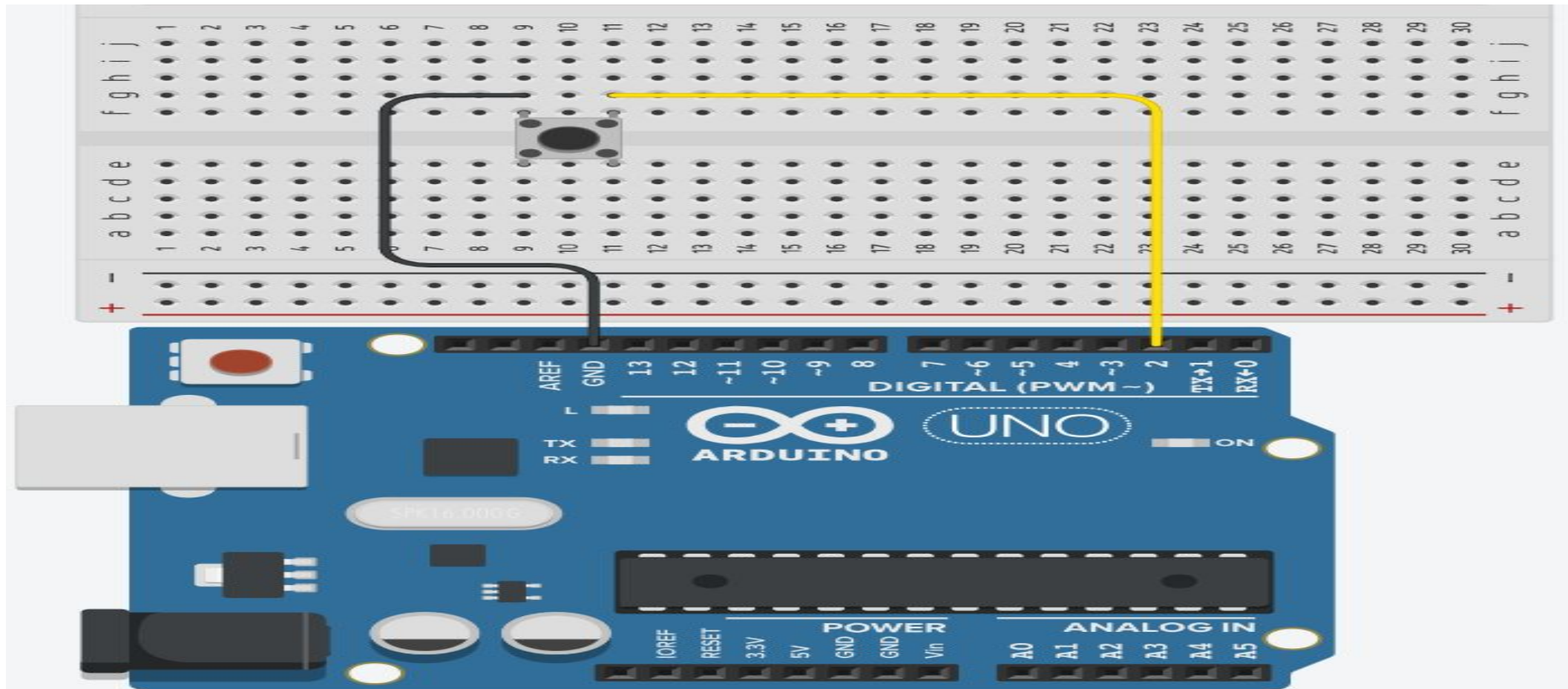
Programando com Arduino

Simulando sensores digitais e seu código - PULLUP:



Programando com Arduino

Simulando sensores digitais e seu código - PULLUP INTERNO:



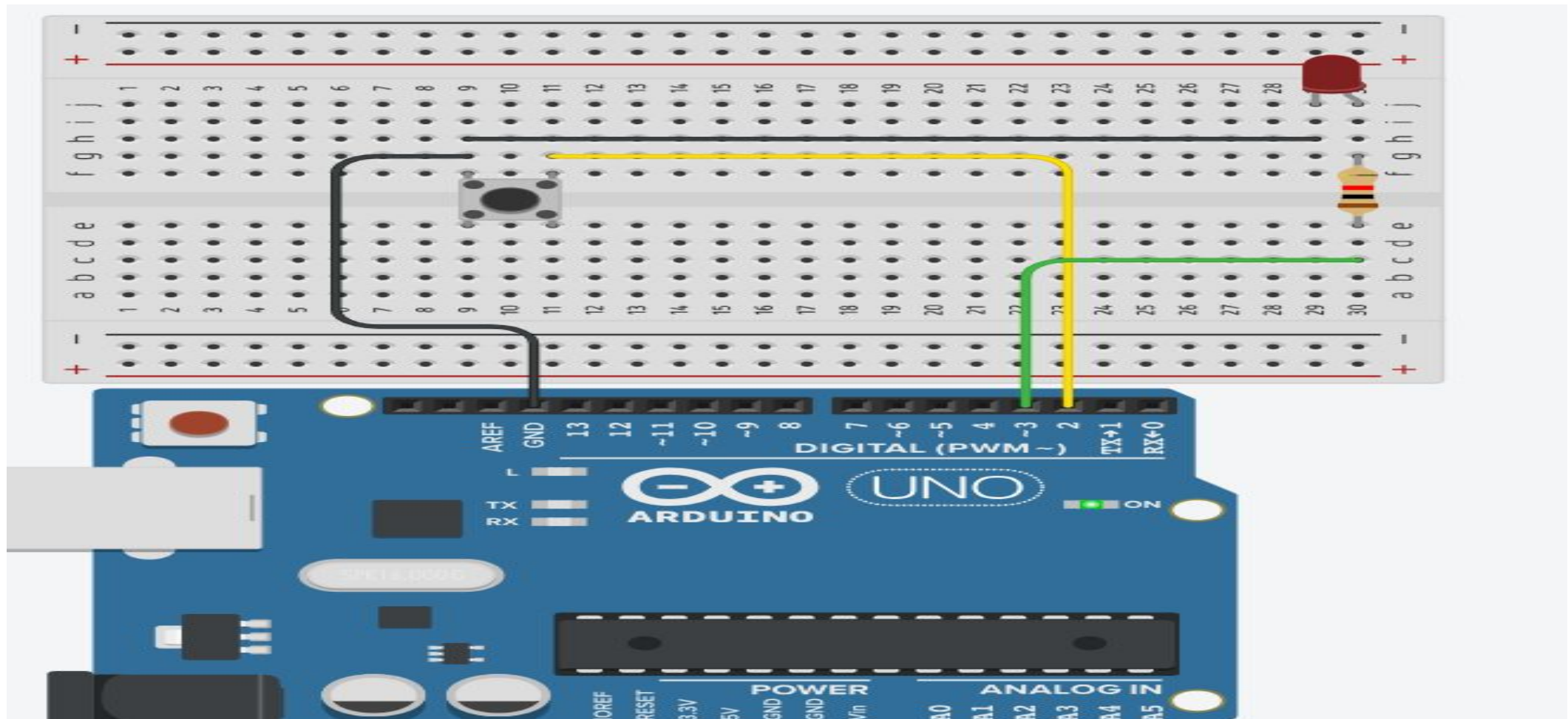
Programando com Arduino

Leitura de taxas no monitor serial:



Programando com Arduino

Aplicação:



OBRIGADO!