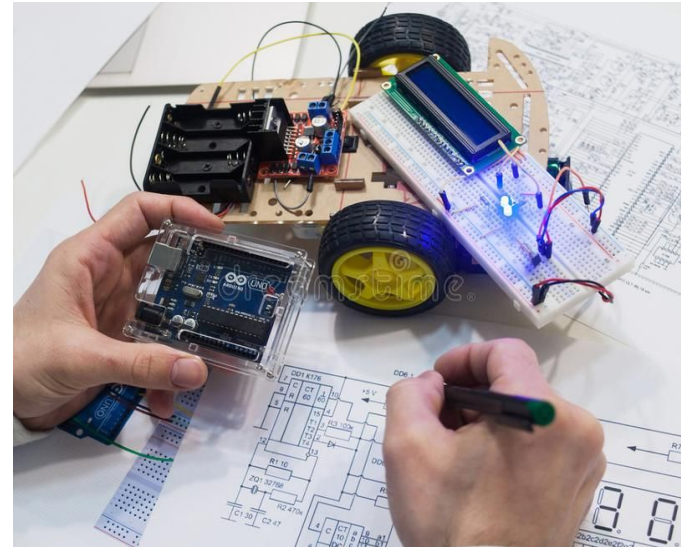
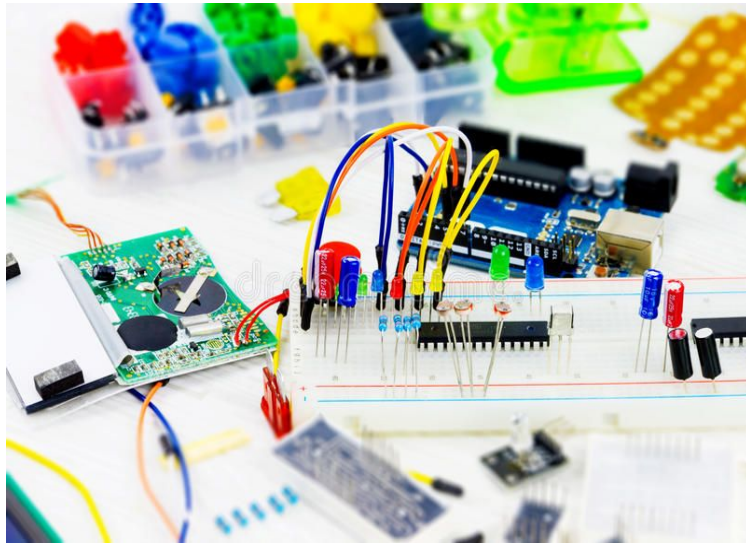


Laboratório de Introdução à Engenharia de Computação

Aula 9 - Comunicação Serial + Saídas Analógicas



Comunicação Serial

Comunicação Serial

- A comunicação serial é amplamente utilizada para **comunicar o Arduino com outros dispositivos** como PC's, Tablets, Bluetooth entre outros.
- Principais funções:
 - `available()` -> Obtém o número de bytes disponíveis para leitura na porta serial.
 - `begin()` -> Configura a taxa de dados para transmissão serial de dados
 - `print()` -> Imprime dados na porta serial em formato legível por humanos (texto ASCII)
 - `println()` -> Idem a `print()`, porém adicionando uma nova linha
 - `read()` -> Lê dados a partir da porta serial.

Primeiro código usando Serial

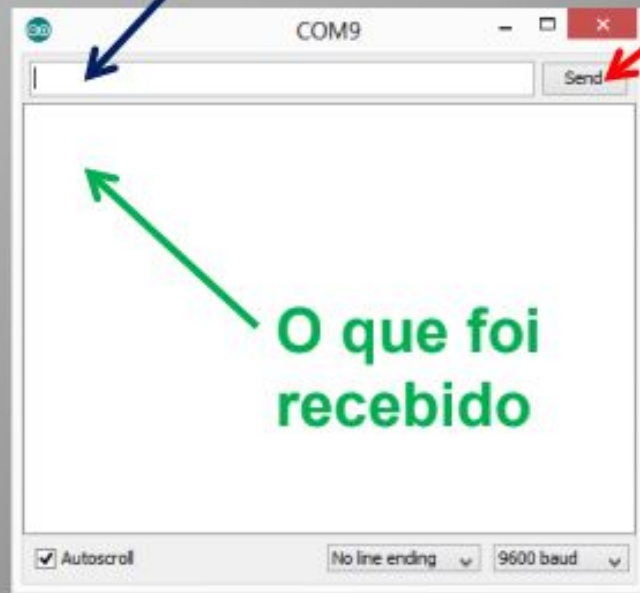
```
1
2 char entrada;
3
4 void setup()
5 {
6     Serial.begin(9600);
7 }
8
9 void loop()
10 {
11     if (Serial.available() > 0){
12         entrada = Serial.read();
13         Serial.print(entrada);
14     }
15 }
16
17
18 |
```

- ▶ Carregar o programa no arduino
- ▶ Executar a comunicação serial

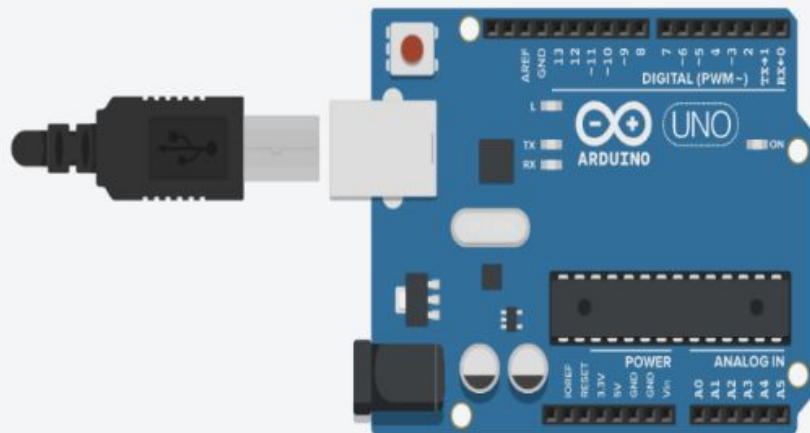


O que será enviado

enviar



No Tinkercad:



Abrir comunicação serial

Texto

```
1 char entrada;  
2 void setup()  
3 {  
4   Serial.begin(9600);  
5 }  
6  
7 void loop()  
8 {  
9   if (Serial.available() > 0) {  
10    entrada = Serial.read();  
11    Serial.print (entrada);  
12  }  
13 }  
14
```

Monitor serial

Os dados recebidos aparecem aqui

Dados a serem enviados aqui

Env.

Apag.

enviar

Testando o Serial:

```
2  int numero = 0;
3
4  void setup()
5  {
6      Serial.begin(9600);
7  }
8
9  void loop()
10 {
11     Serial.print("Valor:");
12     Serial.println(numero);
13     delay(500);
14     numero = numero + 1;
15 }
16
17
18
```



Monitor serial

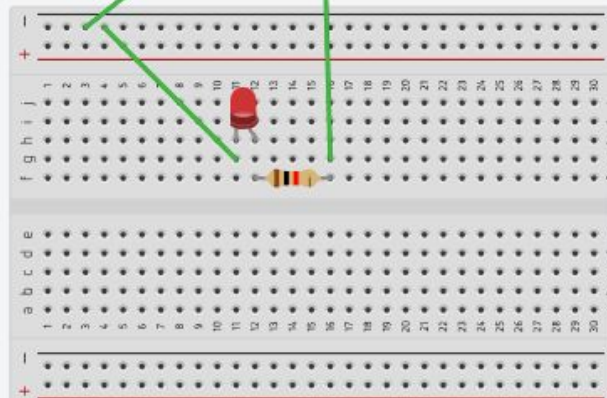
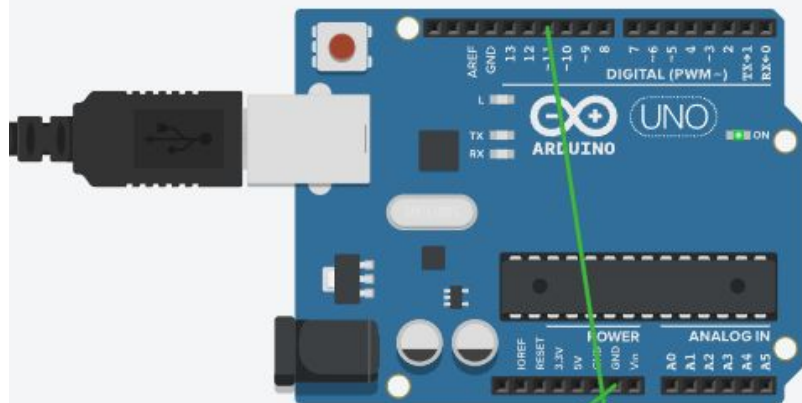
Valor:30
Valor:31
Valor:32
Valor:33
Valor:34
Valor:35
Valor:36
Valor:37

Escrevendo números

```
2 int numero;  
3  
4 void setup()  
5 {  
6   Serial.begin(9600);  
7 }  
8  
9 void loop()  
10 {  
11   if (Serial.available() > 0){  
12     numero = Serial.read();  
13     Serial.println(numero);  
14     delay(500);  
15   }  
16 }
```

X

```
2 int numero;  
3  
4 void setup()  
5 {  
6   Serial.begin(9600);  
7 }  
8  
9 void loop()  
10 {  
11   if (Serial.available() > 0){  
12     numero = Serial.read();  
13     Serial.println(numero - 48);  
14     delay(500);  
15   }  
16 }
```

```
1
2 char entrada = '0';
3 int led = 11;
4
5 void setup()
6 {
7     Serial.begin(9600);
8     pinMode(led, OUTPUT);
9 }
10
11 void loop()
12 {
13     if (Serial.available() > 0){
14         entrada = Serial.read();
15         if (entrada == 'a')
16             digitalWrite(led, 1);
17         if (entrada == 'b')
18             digitalWrite(led, 0);
19     }
20 }
21
22
23
```



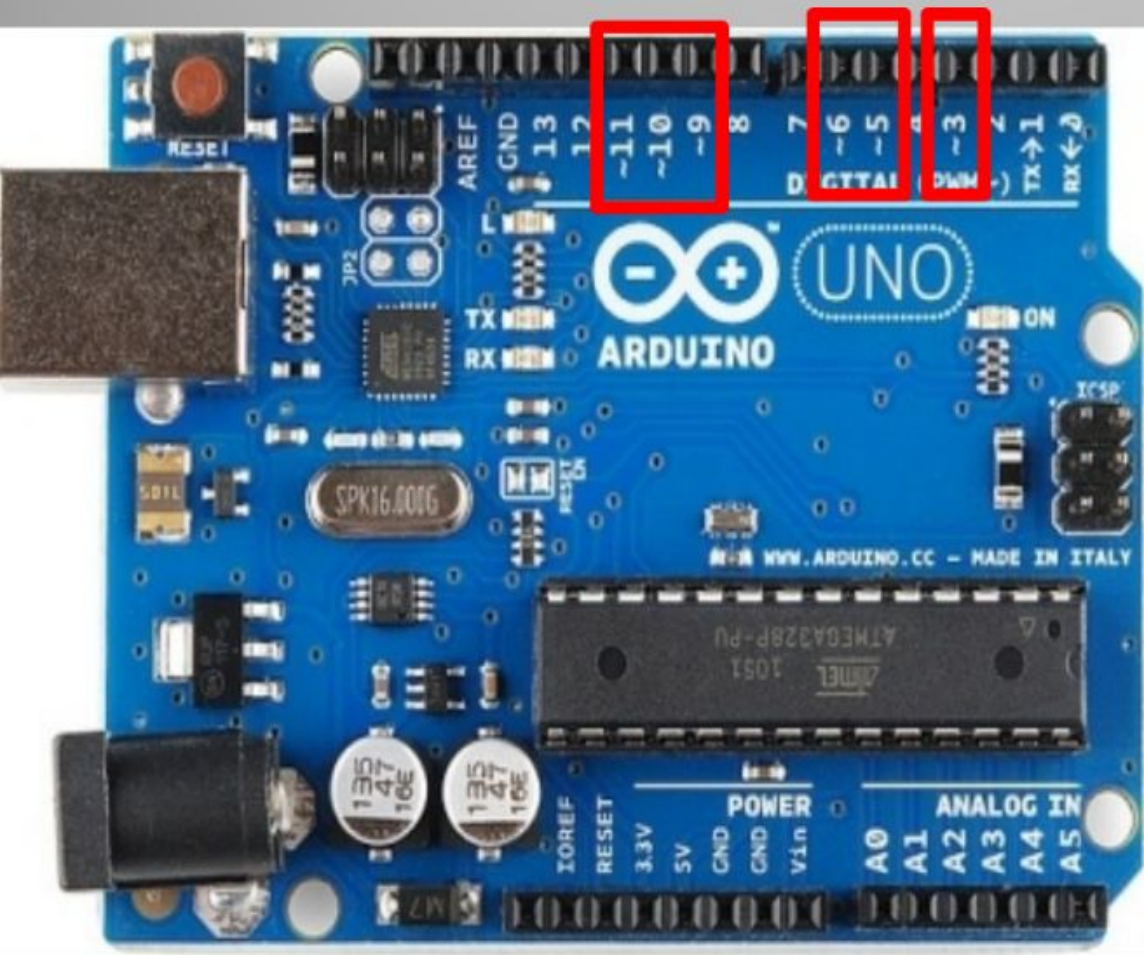
Monitor serial

Saídas Analógicas

Saída analógica

Para o controle do brilho de um Led, usaremos uma saída de forma analógica, ou seja, iremos dividir o intervalo de 0 a 5 volts em 256 valores e iremos aplicar esses valores ao Led.





PINOS com PWM:

~ 3,

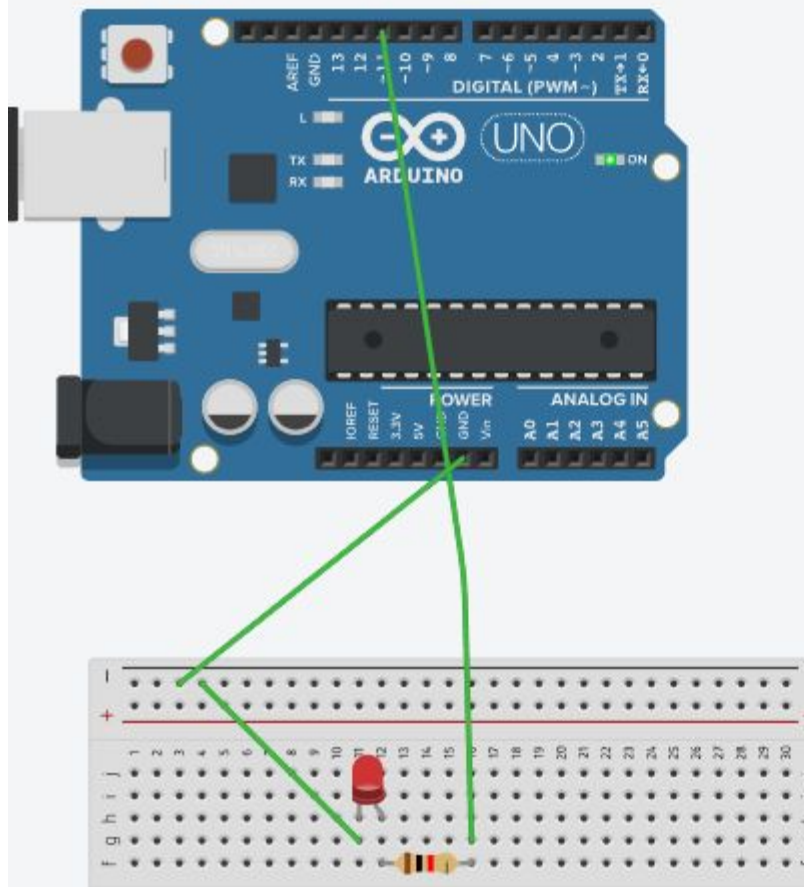
~ 5,

~ 6,

~ 9,

~ 10,

~ 11



```
1
2 int led = 11;
3
4 void setup()
5 {
6   Serial.begin(9600);
7   pinMode(led, OUTPUT);
8 }
9
10 void loop()
11 {
12   analogWrite(led, 0);
13   delay(200);
14   analogWrite(led, 31);
15   delay(200);
16   analogWrite(led, 63);
17   delay(200);
18   analogWrite(led, 127);
19   delay(200);
20   analogWrite(led, 191);
21   delay(200);
22   analogWrite(led, 255);
23   delay(200);
24
25 }
26
27
28
```

Agora é com você!