Medindo distâncias com Arduíno

A equipe que pisca/grita

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

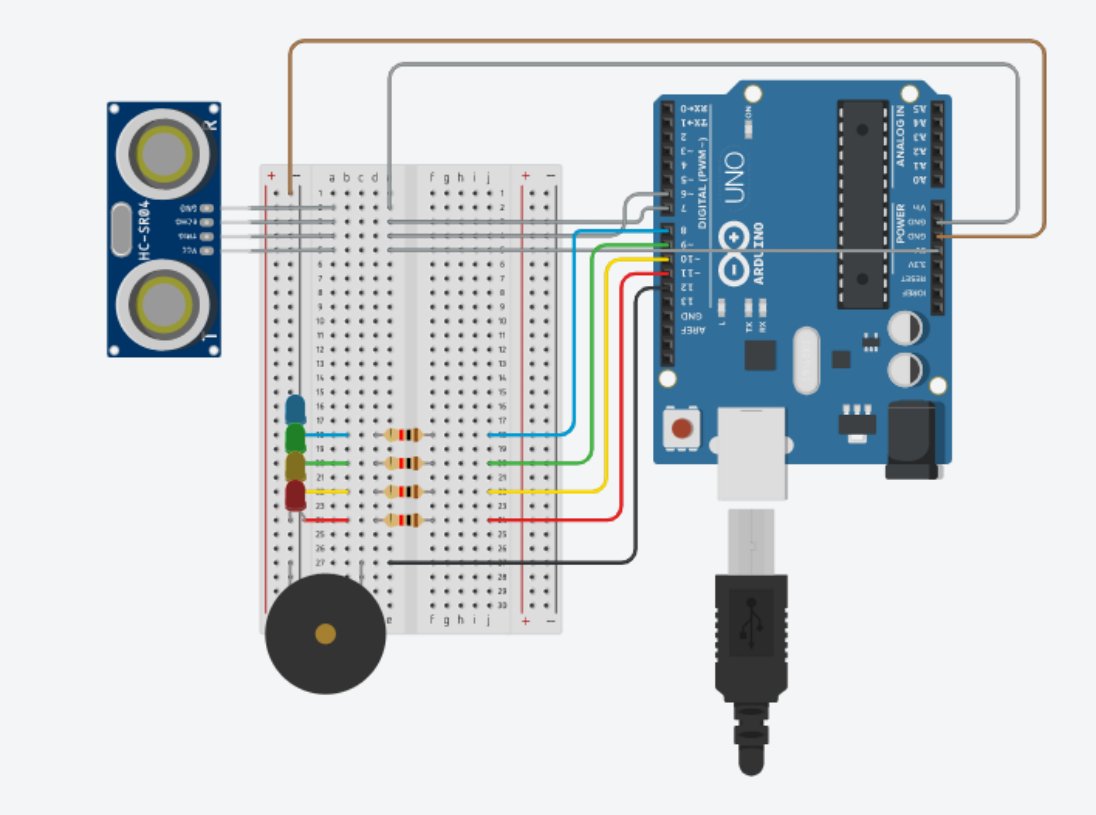
Placas de Prototipagem: Arduíno UNO REV3

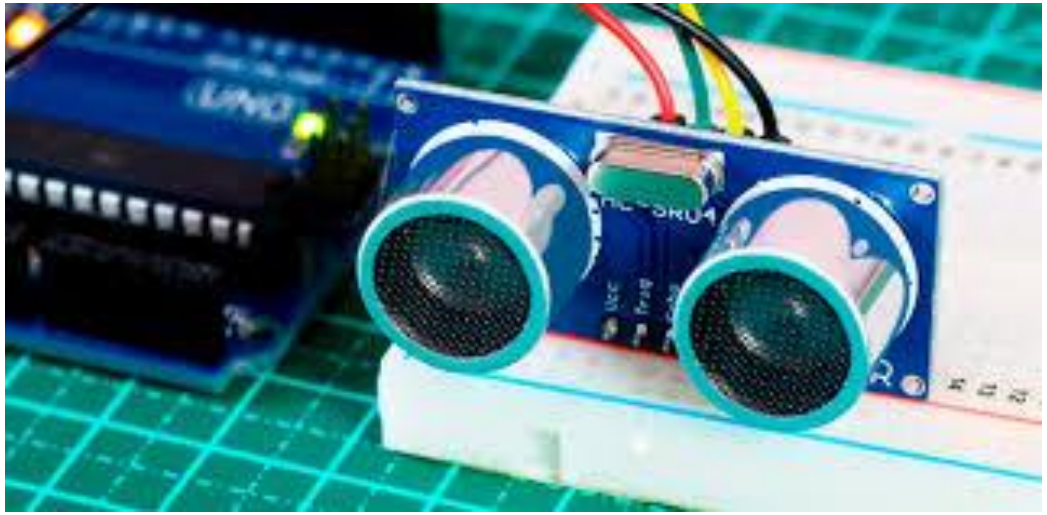
- O que são?
- Como funcionam?

UFPR 105 ANOS DE ORGULHO

Módulo: HC-SR04

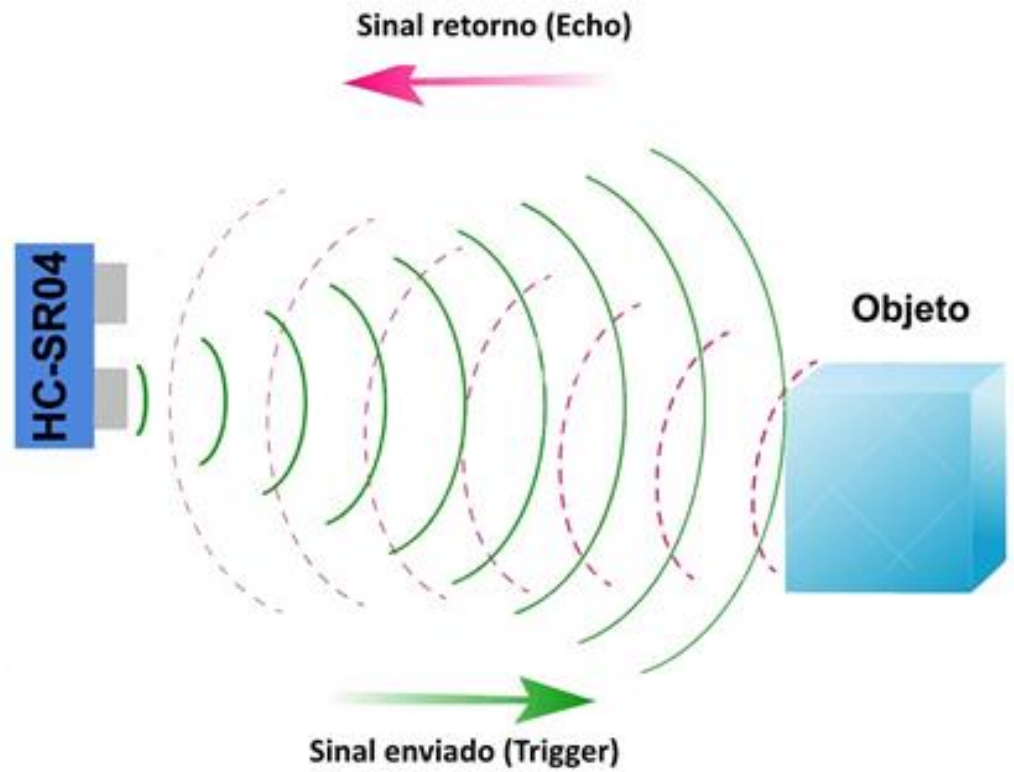
- O que é?

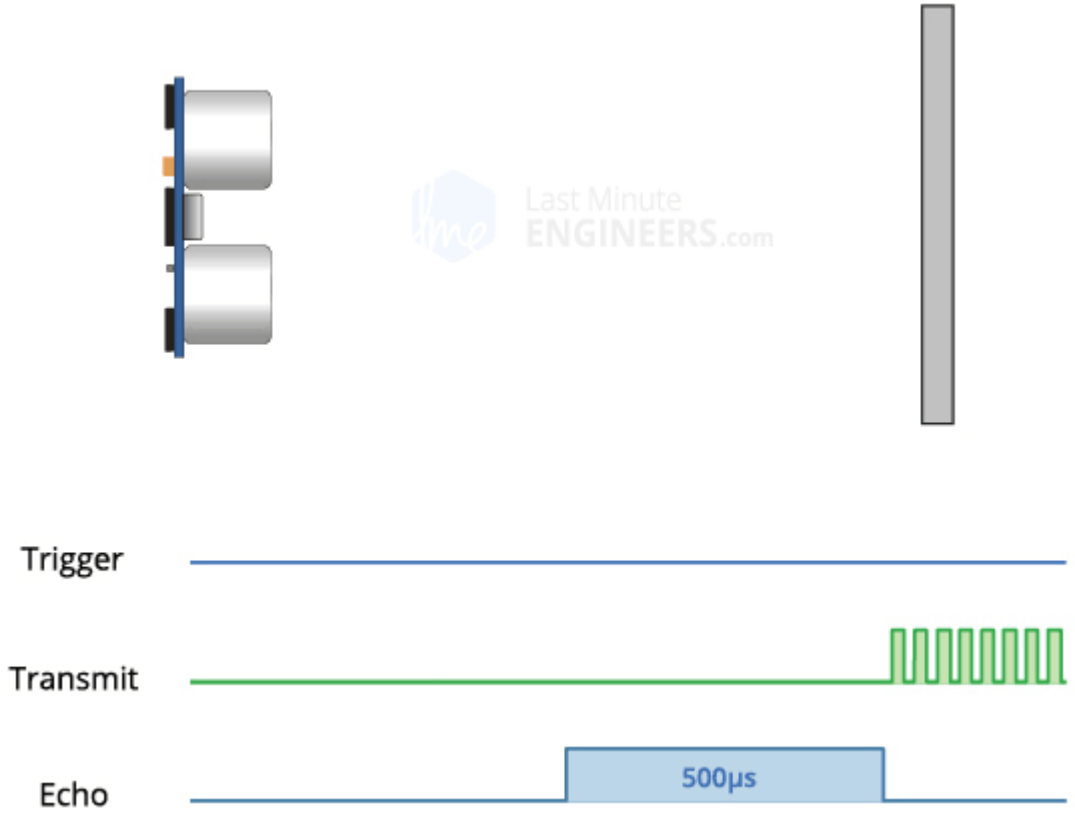




Módulo: HC-SR04

- Como é feita a medição





Tradução para Python

- Para melhor visualização e compreensão do código na linguagem C++, segue o código na linguagem figurativa em Python.

```
58  #INÍCIO DO CÓDIGO PARA O USUÁRIO
59
60  #Usuário informa se deseja ligar, ou não o sensor.
61  ard = input("(L)iga ou (D)eslige o Sensor:").upper()
62
63  #Enquanto o usuário informar "D", o sensor permanecerá desligado.
64  while ard != "L":
65      print("Sensor Desligado, ligue o Sensor.")
66      ard = input("Liga ou Desliga o Sensor:").upper()
```

```
68  #Quando o usuário informar que deseja Ligar o sensor, o mesmo irá ligar.
69  if ard == "L":
70      print("Sensor ligado")
71      #Configure os LED's ou o sensor apenas uma vez
72      parametro = input("Deseja configurar LED ou Sensor?").upper()
73      while parametro != "LED" and parametro != "SENSOR":
74          #Caso algo diferente de LED e SENSOR seja digitado
75          print("Parametro Inexistente")
76          parametro = input("Deseja configurar LED ou Sensor?").upper()
77      #O while acima impossibilita que o parâmetro seja diferente de "LED" e "SENSOR".
78      #Se o parâmetro for "LED" o programa pedirá a próxima configuração, que só poderá ser SERSOR
```

```
#Leitura de distância em Arduinos


 #Função responsável por configurar as saídas de luzes dos LED's
#E definir o emissor/captor de sinais no arduino.
def pinMode(parametro):
    if parametro == "LED":
        status = "Todos configurados como Output"
    else:
        status = "É Output/Input, Pino_echo é Entrada e Pino_trigger é Saída"
    return status
```

```
81      #Caso contrário, o programa pedirá novamente o parâmetro.
82      if parametro == "LED":
83          print(pinMode(parametro))
84          parametro = input("Deseja configurar LED ou Sensor?").upper()
85          while parametro != "SENSOR":
86              if parametro == "LED":
87                  print("LED já configurado!")
88              else:
89                  print("Parametro Inexistente")
90              parametro = input("Deseja configurar LED ou Sensor?").upper()
91          print(pinMode(parametro))
92      #Se o parâmetro for "SENSOR" o programa pedirá a próxima configuração, que só poderá ser LED
93      #Caso contrário, o programa pedirá novamente o parâmetro.
94      else:
95          print(pinMode(parametro))
96          parametro = input("Deseja configurar LED ou Sensor?").upper()
97          while parametro != "LED":
98              if parametro == "SENSOR":
99                  print("SENSOR já configurado!")
100             else:
101                 print("Parametro Inexistente")
102             parametro = input("Deseja configurar LED ou Sensor?").upper()
103         print(pinMode(parametro))
104
```

```
108     #Se a distancia for maior que 0, o programa fara os comandos enviados a função digital_write().
109     while distancia > 0:
110         status_LED, status_Buzzer = digital_write(distancia)
111         if distancia <= 8 and distancia > 0:
112             for i in range(len(status_LED)):
113                 print(status_LED[i])
114         else:
115             print(status_LED)
116         print(status_Buzzer)
117 
118         distancia = int(input("Digite sua distância em Cm do Sensor:"))
```

```
12      #Função responsável por analisar a distância do objeto ao sensor
13      # Acender os LED's e aumentar/diminuir a intensidade do som do Buzzer.
14      def digital_write(dist):
15          print("Sensor sempre é 1")  #Esta sempre ligado.
16
17          #Para cada distância informada pelo usuário, uma ação será realizada.
18          #Distância sempre informada em cm.
19          if dist > 200:
20              status_LED  = "Nenhum LED Ligado"
21              status_Buzzer = "Buzzer Desativado"
22
23          elif dist > 150 and dist <= 200:
24              status_LED  = "LED Azul Ligado"
25              status_Buzzer = "Buzzer com 1/2 segundo"
26
27          elif dist <= 150 and dist > 120:
28              status_LED = "LED Verde Ligado"
29              status_Buzzer = "Buzzer com 1/3 segundo"
30
31          elif dist <= 120 and dist > 80:
32              status_LED  = "LED Verde Ligado"
33              status_Buzzer = "Buzzer com 1/4 segundo"
34
35          elif dist <= 80 and dist > 50:
36              status_LED  = "LED Amarelo Ligado"
37              status_Buzzer = "Buzzer com 1/5 segundo"
```

UFPR 105 ANOS DE ORGULHO

```
39      elif dist <= 50 and dist > 30:
40          status_LED = "LED Amarelo Ligado"
41          status_Buzzer = "Buzzer com 1/6 segundo"
42
43      elif dist <= 30 and dist > 15:
44          status_LED  = "LED Vermelho Ligado"
45          status_Buzzer = "Buzzer com 1/7 segundo"
46
47      elif dist <= 15 and dist > 8:
48          status_LED  = "LED Vermelho Ligado"
49          status_Buzzer = "Buzzer com 1/10 segundo"
50
51      elif dist <= 8 and dist > 0:
52          status_LED = ["LED Azul Ligado", "LED Verde Ligado","LED Amarelo Ligado","LED Vermelho Ligado"]
53          status_Buzzer = "Buzzer com 1/20 Segundo"
54
55      # Retorna o LED que será aceso e a intensidade de som do Buzzer.
56      return status_LED, status_Buzzer
```

```
120     #Caso a distância seja menor ou igual a 0.
121     #O programa entende que o usuário bateu no sensor, e automaticamente encerra.
122     if distancia <= 0:
123         print("Você Bateu, a gente avisou!!!")
124         #PROGRAMA ENCERRADO
```

Código em C++

- A biblioteca utilizada é do sensor ultrassônico;
- As funções usadas, da biblioteca, são:
  - Ultrasonic ultrasonic(trigger, echo)

    Responsável por mostar as portas dos pinos Echo e Trigger do sensor HC-SR04

```
#define pino_trigger 6
#define pino_echo 7
Ultrasonic ultrasonic(pino_trigger, pino_echo);
```

Código em C++

- Ultrasonic.timing()

  Responsável por traduzir as onde de emissão e recepção em tempo.

```
long microsec = ultrasonic.timing();
```

- Ultrasonic.convert(ultrasonic.timing, ultrasonic::CM)

  Responsável por transformar o tempo obtido em centímetros ou polegadas.

```
distancia = ultrasonic.convert(microsec, Ultrasonic::CM);
```

Outras aplicações

- Usando um Arduíno e o módulo HC-SR04, é possível encontrar outras aplicabilidades, como sinaleiros, bebedouros automáticos, radares, sensor de estacionamento, etc.

Flávia Maria Schütz GRR20195838

Izabella Calais Fernandes SO20190290

Karen da Luz Guarnieri GRR20195848

Thiago Batista dos Santos Martins GRR20195874

Dúvidas?

UFPR 105 ANOS DE ORGULHO