

Sensor ultrassónico para medição de ondas



Diagrama ilustrativo da montagem interna:

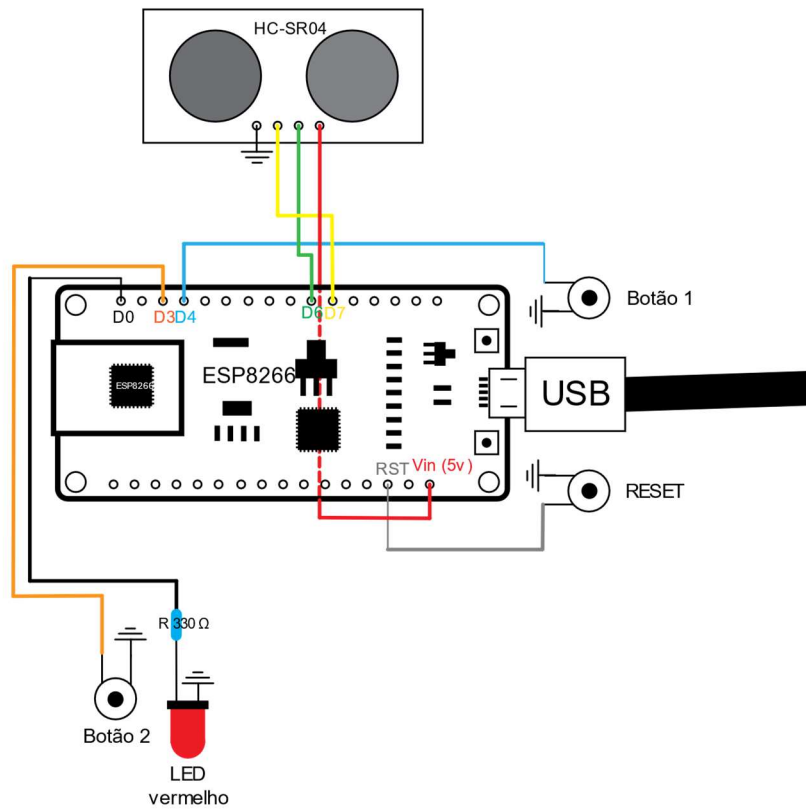


Ilustração externa:

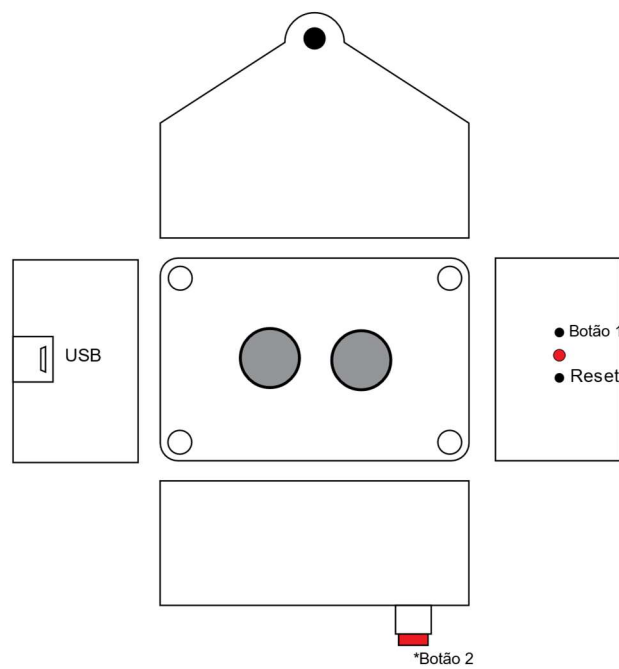


Tabela de componentes:

QUANTIDADE	TIPO	MODELO
1	Microcontrolador	ESP8266
1	Sensor ultrassónico	HC-SR04
1	LED vermelho	Genérico
1	Resistência (330 Ω)	Genérico
3	Botão momentâneo	Genérico
1	Cabo Micro-USB para USB 3m	Genérico

Instruções de uso:

1. Colocar o módulo numa posição acima do sítio onde se pretende efetuar a medida.
2. Ligar o cabo USB a um computador.
3. Abrir um programa com capacidade de monitor de série (por ex. Arduino IDE):
 - a. Selecionar a taxa de transmissão recebida a: 9600
4. Seguir as instruções que serão mostradas no ecrã:
 - a. Inserir a temperatura ambiente e enviar
5. Começar uma contagem:
 - a. Usando o Botão 1, Botão 2 ou enviando "S" pelo monitor de série começa uma contagem com 40ms de intervalo entre cada medição.
 - b. Enviando "N" pelo monitor de série é possível começar uma nova medição com o intervalo entre amostras parametrizável:
 - i. Envie pelo monitor de série o intervalo que pretende (>40ms)
6. Parar uma contagem: Usando o Botão 1, Botão 2 ou enviando "S" pelo monitor de série.
7. Copiar os dados do monitor de série e guardar num ficheiro a escolha, para tratamento dos mesmos posteriormente.

Observação:

1. Se for necessário fazer *reset* ao microcontrolador pode-se enviar "R" pelo monitor de série ou clicando no botão *reset*.
2. Para uso com Arduino ide: Arduino IDE -> Ferramentas -> Monitor

Observações finais:

O módulo é composto por um sensor ultrassônico “HC-SR04” que mede a distância de qualquer objeto até ao mesmo. Tem uma resolução de 0.3 Cm o que faz com que cada medida tenha uma incerteza de $\pm 0.3\text{Cm}$, tem alcance de 2cm até 400cm.

O sensor é composto por um transmissor e recetor, transmite uma onda com frequência de 40 kHz. Quando transmite a onda a onda é refletida e a mesma demora mais ou menos tempo dependendo da distância a que está o objeto. O sensor contém 4 conexões, sendo duas das mesmas de energia, e as duas de comunicação, um desses portos é denominada por “trigger” e outro é denominado de “echo”. Para ser efetuada uma medida o porto “trigger” tem de ser enviado um sinal alto durante 10 μs , o módulo é então assim sinalizado que se pretende fazer uma medição. De seguida envia 8 sinais de 40kHz, através do envio de 8 sinais consegue minimizar o erro devido a interferências externas. Se nenhum dos sinais for obtido no transmissor o sensor envia um sinal alto através do porto “Echo” durante 38ms que sinaliza que a leitura não foi efetuada corretamente (Time-out). Se o sensor receber os sinais transmitidos coloca o porto “Echo” em sinal alto durante o tempo que o sinal demorou desde que foi transmitido até ser recebido. Usando esse tempo, podemos calcular a distância de um objeto relativamente ao sensor através da velocidade do som.

$$\text{Distância}(cm) = \text{VelocidadedoSom}(cm/\mu s) \times \frac{\text{DuraçãodoSinalEcho}(\mu s)}{2}$$

$$\text{Distância} = 0.034 \times \frac{\text{Echo}}{2}$$

Visto que a velocidade do som está relacionada com a densidade do material de propagação, e a densidade está por sua vez relacionada com a temperatura a que se encontra esse material, a equação teve de ser ajustada tendo em conta a temperatura ambiente.

$$\text{Distância} = \left(0.034 \times \frac{\text{Echo}}{2}\right) \times \sqrt{\frac{1 + T}{273.15}} *$$

**Fórmula utilizada por recomendação dos criadores da livraria “NewPing”*

Para operar o sensor foi usado um microcontrolador, neste caso baseado no microchip ESP8266. A tarefa deste microcontrolador coordenar os pedidos de leitura ao sensor e de seguida processar os dados que o sensor transmite para a distância em centímetros usando a fórmula acima.

O microcontrolador foi programado usando programação arduino. No código foi usada uma livreria “NewPing”, escrita por “Tim Eckel“, esta livreria foi escrita para agilizar o uso deste tipo de sensores sendo esta uma forma mais fácil e intuitiva de interagir com o sensor através do micro controlador. A livreria agilizou o desenvolvimento da interface e a mitigação de erros de medida pelo processamento de dados.

Links uteis:

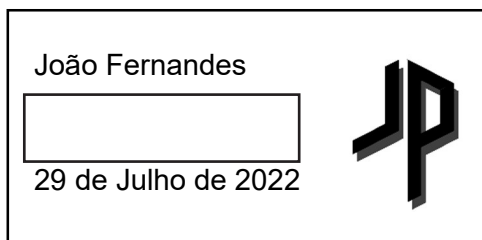
<https://www.arduino.cc/reference/en/libraries/newping/>

Documentação:

<https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home>

Em anexo está o código que foi escrito no microcontrolador à data.

Corresponde á Versão 5 do programa “Sensor ultrassónico para medição de ondas” programado em Arduino.



```

1 //JOAO FERNANDES 2022
2 //SENSOR ULTRASONICO P
3
4
5 #include <NewPing.h>
6 #define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
7 #define ECHO_PIN 13 // Arduino pin tied to echo pin on the ultrasonic sensor.
8 #define MAX_DISTANCE 170 // Maximum distance we want to ping for (in centimeters).
9
10 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and
    maximum distance.
11
12 float temp = 25; // Temperature in Celsius (this value would probably come from a
    temperature sensor).
13 float factor = sqrt(1 + temp / 273.15) / 60.368; // Speed of sound calculation based
    on temperature. //default value 25°
14 short int Mode = 0;
15 bool btnState;
16 unsigned long lastButtonPress = 0;
17 short int btnDelay = 250; //delay for debouncing
18 short int readingN = 0;
19 unsigned long ReadingStart = 0;
20 String READING;
21 String mydata;
22 float avg = 0;
23 String inst = "\nJoão Fernandes 2022\n Para começar leituras clique no botão 1, botão
    2 ou insira 'S' no comando de série.\n Para parar clique no botão 1, botão 2 ou
    insira 'S' no comando de série.\n Para fazer uma leitura com um numero diferente de
    tempo entre amostras (default = 40ms) envie 'N'\n Para RESET envie 'R'";
24 String Code_version = "V5.0";
25 short int Rate = 40; //default
26
27 void setup() {
28     Serial.begin(9600);
29     pinMode(16, OUTPUT);
30     pinMode(0, INPUT_PULLUP);
31     pinMode(4, INPUT_PULLUP);
32
33     digitalWrite(16, HIGH);
34     delay(100);
35     digitalWrite(16, LOW);
36     delay(500);
37     digitalWrite(16, HIGH);
38     delay(100);
39     digitalWrite(16, LOW);
40     delay(1000);
41     Serial.print("\nPROGRAMA A INICIAR \n");
42     Serial.print("\nInsira a temperatura ambiente: ");
43
44     // while(!Serial.available()) {
45     while (!Serial || !Serial.available()) {
46
47         if (millis() % 250 == 0) {
48             digitalWrite(16, HIGH);
49         }
50         else if (millis() % 100 == 0) {
51             digitalWrite(16, LOW);
52         }
53         if (millis() % 3000 == 0) {
54             Serial.print("\nInsira a temperatura ambiente: ");
55             Serial.flush();
56         }
57         yield();
58     }
59
60
61     while (mydata.toInt() == 0) {
62         mydata = Serial.readStringUntil('\n');
63         Serial.println(mydata);
64         Serial.println("Temperatura guardada\n==== Programa Iniciado =====\n");
65         Serial.flush();
66         yield();
67     }

```

```

68     temp = mydata.toInt(); // Temperature in Celsius (this value would probably come
69     factor = sqrt(1 + temp / 273.15) / 60.368; // Speed of sound calculation based on
        temperature. //default value 25°
70
71     for (int i = 2; i > 0; i--) {
72         float readingB = ((float)sonar.ping_median(5) * factor);
73         avg = avg + readingB;
74         READING = "Distancia BASE: " + (String)readingB + " cm ";
75         READING.replace(".", ",");
76         Serial.println(READING);
77         delay(300);
78     }
79     avg = avg / 2;
80     READING = "\nMedia: " + (String)avg + " cm ";
81     READING.replace(".", ",");
82     Serial.println(READING);
83     Serial.print("\n Versão: ");
84     Serial.println(Code_version);
85     Serial.println(inst);
86     Serial.println("*****\n");
87
88     Serial.println("A aguardar inicio -> Botão 1 ou Botão 2 ou escreva 'S' OU 'N' ");
89     Mode = 0;
90 }
91
92 void loop() {
93
94
95     if (Mode == 0) {
96         mydata = "";
97         while (Mode == 0) {
98
99
100             if (millis() % 500 == 0) {
101                 digitalWrite(16, HIGH);
102             }
103             else if (millis() % 250 == 0) {
104                 digitalWrite(16, LOW);
105             }
106
107             if (Serial.available() > 0) {
108                 mydata = Serial.readStringUntil('\n');
109                 Serial.println(mydata);
110
111                 if (mydata == "S") {
112                     Mode = 1;
113                     Serial.flush();
114                 }
115                 else if (mydata == "R") {
116                     ESP.reset();
117                 }
118                 else if (mydata == "N") {
119                     Mode = 2;
120                     Serial.flush();
121                 }
122                 else {
123                     Serial.println("Modo inválido -> 'S' OU 'N'");
124                     mydata = "";
125                     Serial.flush();
126                 }
127             }
128
129             btnState = digitalRead(0); // read input value
130             if (btnState == LOW) {
131                 if (millis() - lastButtonPress > btnDelay) {
132                     Mode = 1;
133                     btnState == HIGH;
134                 }
135                 lastButtonPress = millis();
136             }
137             btnState = digitalRead(4); // read input value
138             if (btnState == LOW) {

```

```

139         if (millis() - lastButtonPress > btnDelay) {
140             Mode = 1;
141             btnState == HIGH;
142         }
143         lastButtonPress = millis();
144     }
145     yield();
146 }
147 }
148
149 else if (Mode == 2) {
150     Rate = 0;
151     avg = 0;
152
153     Serial.print("\nMODO 2\n");
154     Serial.print("\nInsira o numero de ms entre amostras pretendido (minimo 40ms): ");
155
156     while (Rate < 40) {
157         if (Serial.available() > 0) {
158             mydata = Serial.readStringUntil('\n');
159             if (mydata.toInt() > 40 && mydata.toInt() < 5000) {
160                 Rate = mydata.toInt();
161                 Serial.flush();
162             }
163             else if (mydata == "R") {
164                 ESP.reset();
165             }
166             else if (mydata == "S") {
167                 Mode = 0;
168                 Serial.flush();
169                 break;
170             }
171             else {
172                 Serial.println("\nValor Inválido ('R' PARA RESET & 'S' Para voltar ao menu
173                 inicial)\nInsira um valor >40ms");
174                 Serial.flush();
175             }
176             // Serial.println(mydata);
177             // Serial.println(mydata.toInt());
178         }
179         yield();
180     }
181
182     readingN++;
183     if (Mode != 0) {
184         Serial.print("\n***** LEITURA NUMERO: ");
185         Serial.print(readingN);
186         Serial.println(" *****");
187
188         for (int i = 5; i > 0; i--) {
189             float readingB = ((float)sonar.ping_median(5) * factor);
190             avg = avg + readingB;
191             READING = "Distancia BASE: " + (String)readingB + " cm ";
192             READING.replace(".", ",");
193             Serial.println(READING);
194             delay(300);
195         }
196         avg = avg / 5;
197         READING = "\nDistancia BASE: Média = " + (String)avg + " cm ";
198         READING.replace(".", ",");
199         Serial.println(READING);
200         Serial.print("\nDISTÂNCIA (cm) || TEMPO (ms)\n\n");
201
202         ReadingStart = millis();
203     }
204
205     while (Mode != 0) {
206         digitalWrite(16, HIGH);
207
208         if (Serial.available() > 0) {
209             mydata = Serial.readStringUntil('\n');

```



```

210     if (mydata == "S") {
211         Mode = 0;
212         Serial.flush();
213     }
214     if (mydata == "R") {
215         ESP.reset();
216     }
217 }
218
219
220 btnState = digitalRead(0); // read input value
221 if (btnState == LOW) {
222     if (millis() - lastButtonPress > btnDelay) {
223         Mode = 0;
224         btnState == HIGH;
225     }
226     lastButtonPress = millis();
227 }
228 btnState = digitalRead(4); // read input value
229 if (btnState == LOW) {
230     if (millis() - lastButtonPress > btnDelay) {
231         Mode = 0;
232         btnState == HIGH;
233     }
234     lastButtonPress = millis();
235 }
236
237 if (millis() % Rate == 0) {
238
239     READING = String(((float)sonar.ping_median(1) * factor), 2) + " " + (String)(
millis() - ReadingStart) + "\n" ;
240     READING.replace(".", ",");
241     Serial.print(READING);
242     Serial.flush();
243 }
244
245     yield();
246 }
247 }
248
249 else {
250     avg = 0;
251     readingN++;
252     Serial.print("\n***** LEITURA NUMERO: " );
253     Serial.print(readingN );
254     Serial.println(" *****");
255     // Serial.print("BASE: ");
256     // Serial.print(avg);
257     for (int i = 5; i > 0; i--) {
258         float readingB = ((float)sonar.ping_median(5) * factor);
259         avg = avg + readingB;
260         READING = "Distancia BASE: " + (String)readingB + " cm ";
261         READING.replace(".", ",");
262         Serial.println(READING);
263         delay(300);
264     }
265     avg = avg / 5;
266     READING = "\nDistancia BASE: Média = " + (String)avg + " cm ";
267     READING.replace(".", ",");
268     Serial.println(READING);
269     Serial.print("\n Intervalo de tempo entre amostras: 40ms \n\n DISTÂNCIA (cm) ||
TEMPO (ms)\n\n");
270
271     ReadingStart = millis();
272
273     while (Mode != 0) {
274
275
276         digitalWrite(16, HIGH);
277
278         if (Serial.available() > 0) {
279             mydata = Serial.readStringUntil('\n');
280

```

```

281     if (mydata == "S") {
282         Mode = 0;
283         Serial.flush();
284     }
285     else if (mydata == "R") {
286         ESP.reset();
287     }
288     else {
289         Serial.flush();
290     }
291 }
292
293
294 btnState = digitalRead(0); // read input value
295 if (btnState == LOW) {
296     if (millis() - lastButtonPress > btnDelay) {
297         Mode = 0;
298         btnState == HIGH;
299     }
300     lastButtonPress = millis();
301 }
302 btnState = digitalRead(4); // read input value
303 if (btnState == LOW) {
304     if (millis() - lastButtonPress > btnDelay) {
305         Mode = 0;
306         btnState == HIGH;
307     }
308     lastButtonPress = millis();
309 }
310
311 if (millis() % 40 == 0) {
312
313     READING = String(((float)sonar.ping_median(1) * factor), 2) + " " + (String)(
314         millis() - ReadingStart) + "\n" ;
315     READING.replace(".", ",");
316     Serial.print(READING);
317     Serial.flush();
318 }
319 yield();
320 }
321 }
322
323 }
324
325 //JOAO FERNANDES 2022
326

```