



***FACULDADE DE TECNOLOGIA SANTO ANDRÉ***

***Integrantes***

***Rubia Archila Lima De Almeida***

***Gustavo Sasaki Rodrigues***

***Pedro Henrique Silva Friolani***

***SENSOR DE RÉ***

***Santo André***

***2019***

## ***SUMÁRIO***

<b>ANALISE .....</b>	<b>3</b>
<b>SENSOR ULTRASSÓNICO .....</b>	<b>3</b>
<b>LISTA DE MATERIAIS .....</b>	<b>5</b>
<b>LISTA DE ENTRADAS E SAÍDAS.....</b>	<b>5</b>
<b>ANEXOS .....</b>	<b>6</b>
<b>CIRCUITO.....</b>	<b>6</b>
<b>esquema elétrico .....</b>	<b>7</b>
<b>PROGRAMAÇÃO.....</b>	<b>8</b>

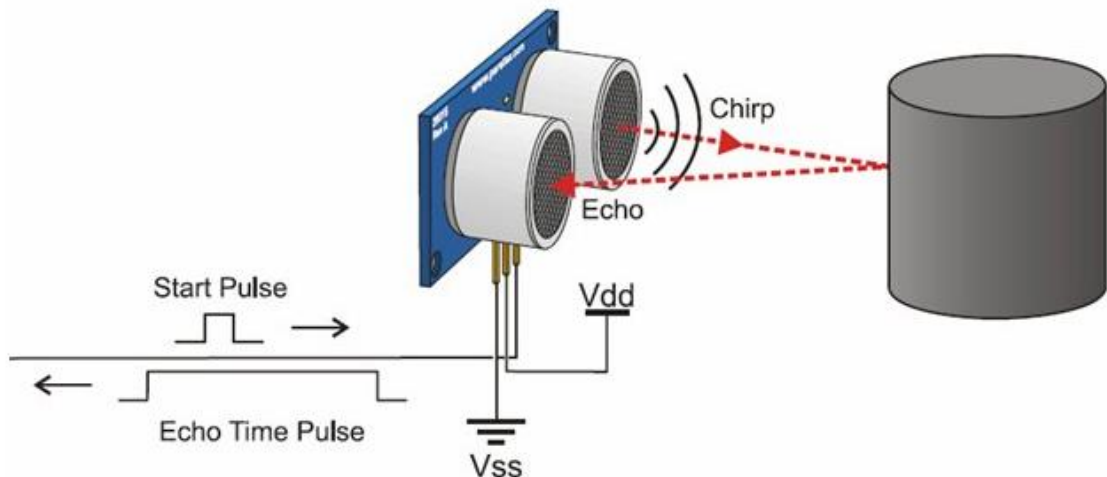
## ANALISE

### Sensor Ultrassônico

*O Sensor de Distância Ultrassônico HC-SR04 é um sensor capaz de medir distâncias de 2 cm à 4 metros. Este sensor possui um emissor e receptor de sinal ultrassônico, possibilitando ao usuário medir distancias de acordo com a reflexão do sinal.*

*O Sensor Ultrassônico trabalha com diversos microcontroladores, como Arduino e PIC. Trabalha com tensão de 5V, tem um consumo de energia de 2mA, além de ter uma precisão de aproximadamente 3mm e o seu sinal ultrassônico viaja a uma frequência de 40kHz, tendo um excelente custo benefício.*

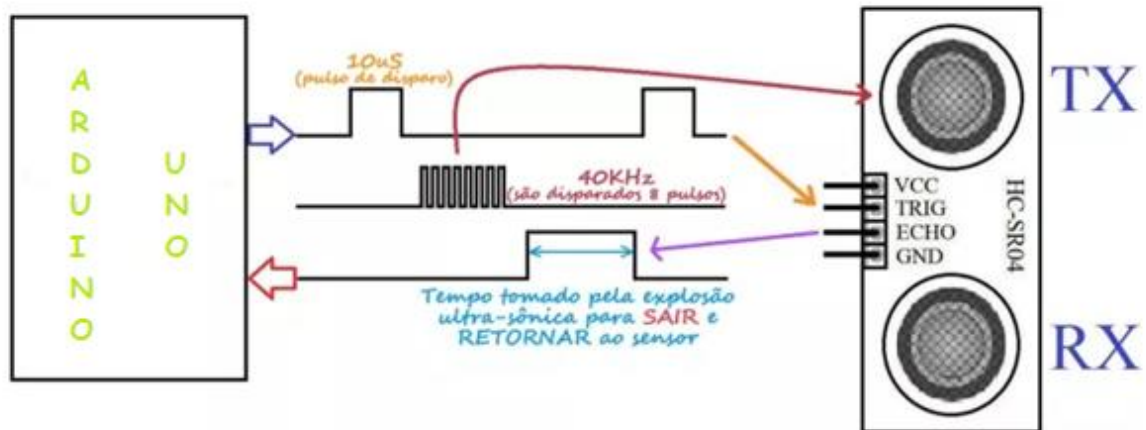
Figura 1-Sensor Ultrasonico



<https://blog.usinainfo.com.br/sensor-ultrassonico-de-distancia-hc-sr04/>

A orientação correta do sensor é fundamental para se obter um bom desempenho do sistema. Nesse caso, especial atenção deve ser tomada com os ângulos para que o objeto passe pela região de maior intensidade do sinal e a reflexão ocorra na direção em que está o sensor.

Figura 2



Fonte: Montagem, autor próprio.

### ***LISTA DE MATERIAIS***

<b><i>N°</i></b>	<b><i>MATERIAL</i></b>	<b><i>QUANTIDADE</i></b>	<b><i>OBSERVAÇÃO</i></b>
<i>1</i>	<i>PROTOBOARD</i>	<i>1</i>	<i>20X7</i>
<i>2</i>	<i>SENSOR</i>	<i>1</i>	<i>HC-SR04</i>
<i>3</i>	<i>LED</i>	<i>5</i>	<i>red,greem,yelow,blue,write</i>
<i>4</i>	<i>ARDUINO</i>	<i>1</i>	<i>uno</i>
<i>5</i>	<i>RESISTOR</i>	<i>5</i>	<i>220 <math>\Omega</math></i>
<i>6</i>	<i>FONTE</i>	<i>1</i>	<i>5v</i>
<i>7</i>	<i>JUMPER</i>	<i><math>\cong 20</math></i>	<i>variados</i>

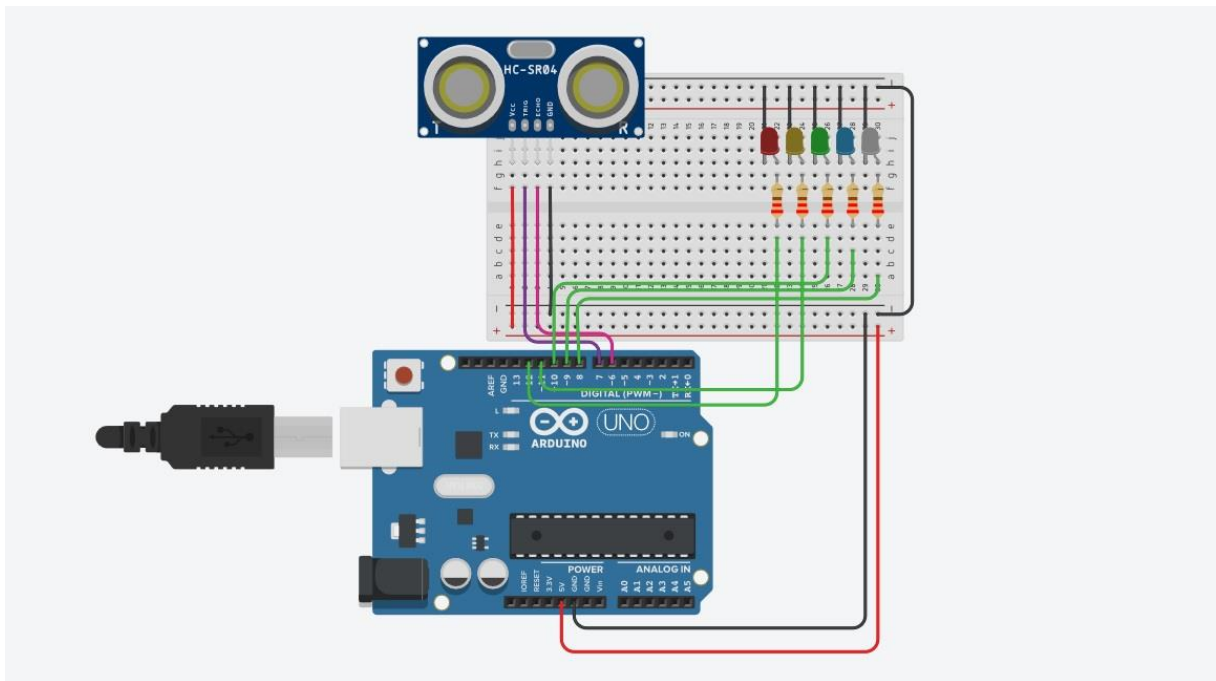
### ***LISTA DE ENTRADAS E SAÍDAS***

LED_BRANCO_PB0	-SAÍDA
LED_AZUL_PB1	- SAÍDA
LED_VERDE_PB2	- SAÍDA
LED_AMARELO_PB3	- SAÍDA
LED_VERMELHO_PB4	- SAÍDA
SENSOR_TRIG_PD7	-SAÍDA
SENSOR_ECHO_PD6	-ENTRADA

## ANEXOS

<https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>

## CIRCUITO



*LED VERMELHO distancia < 10cm*

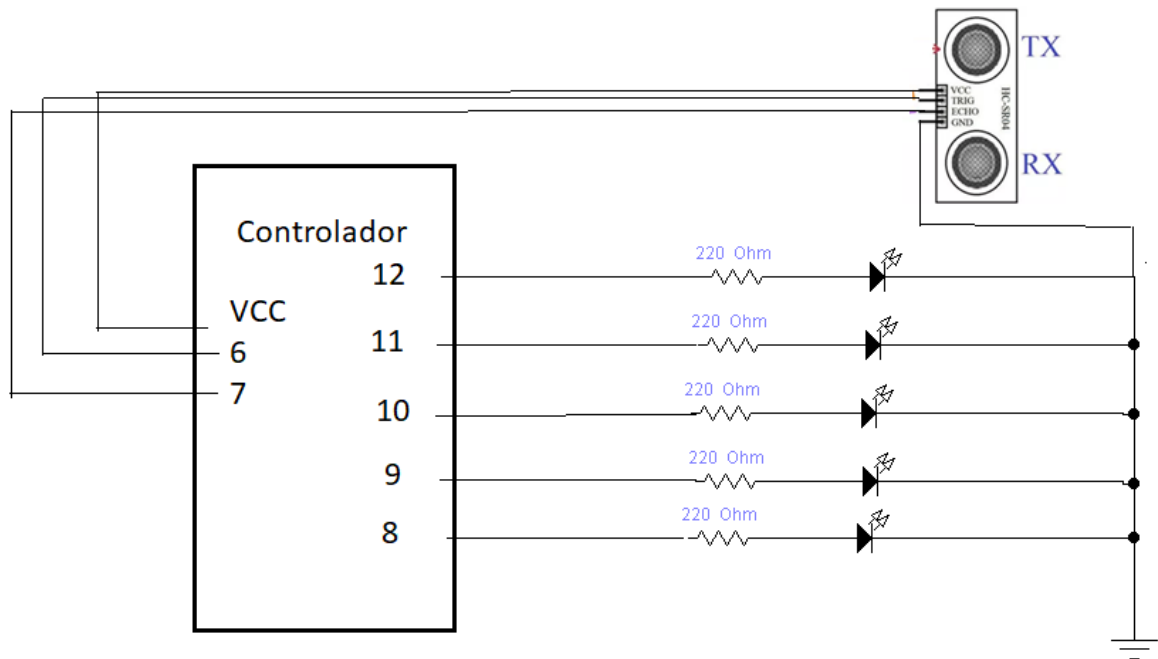
*LED AMARELO distancia < 20cm and distancia >= 10cm*

*LED VERDE distancia < 30cm and distancia >= 20cm*

*LED AZUL distancia < 40cm and distancia >= 30cm*

*LED BRANCO distancia > 40cm*

## ESQUEMA ELÉTRICO



## **PROGRAMAÇÃO**

```

#define trig 7 //pd7
#define echo 6 //pd6
#define led1 8 //branco pb0
#define led2 9 //azul pb1
#define led3 10 //verde pb2
#define led4 11 //amarelo pb3
#define led5 12 //vermelho pb4

long duracao;
float distancia;

#define BAUD 9600
#include <avr/io.h>
#include <stdio.h>
#include "minhasMacros.h"
#include "minhaCom.h"
#include "minhaAdc.h"
#include <util/setbaud.h>

void setup() {
    initUSART();
    setBit(DDRD,PD7);
    setBit(DDRB,PB0);
    setBit(DDRB,PB1);
    setBit(DDRB,PB2);
    setBit(DDRB,PB3);
    setBit(DDRB,PB4);
    clearBit(DDRD,PD6);

```



```

    Serial.begin(9600);
}

void loop() {
    clearBit(PORTD, PD7);
    delayMicroseconds(5);
    setBit(PORTD, PD7);
    delayMicroseconds(10);
    clearBit(PORTD,PD7);

    duracao = pulseIn (echo, HIGH);

    distancia = duracao * 0.017;

    Serial.print("Distancia em cm: ");
    Serial.println(distancia);
    delay (1000);

    leddistancia();

}

void leddistancia(){
    clearBit(DDRB,PB0);
    clearBit(DDRB,PB1);
    clearBit(DDRB,PB2);
    clearBit(DDRB,PB3);
    clearBit(DDRB,PB4);

```

```
if (distancia > 40) //acende led braco
{
    setBit(PORTB,PB0);
    clearBit(PORTB,PB1);
    clearBit(PORTB,PB2);
    clearBit(PORTB,PB3);
    clearBit(PORTB,PB4);
}

else if (distancia < 40 and distancia >=30) //acende led azul
{
    setBit(PORTB,PB1);
    clearBit(PORTB,PB0);
    clearBit(PORTB,PB2);
    clearBit(PORTB,PB3);
    clearBit(PORTB,PB4);
}

else if (distancia < 30 and distancia >=20) //acende led verde
{
    setBit(PORTB,PB2);
    clearBit(PORTB,PB0);
    clearBit(PORTB,PB1);
    clearBit(PORTB,PB3);
    clearBit(PORTB,PB4);
}

else if (distancia < 20 and distancia >=10)//acende led amarelo
{
    setBit(PORTB,PB3);
    clearBit(PORTB,PB0);
    clearBit(PORTB,PB1);
```

```
clearBit(PORTB,PB2);  
clearBit(PORTB,PB4);  
}  
else if (distancia < 10) //acende led vermelho  
{  
setBit(PORTB,PB4);  
clearBit(PORTB,PB0);  
clearBit(PORTB,PB1);  
clearBit(PORTB,PB2);  
clearBit(PORTB,PB3);  
}}
```