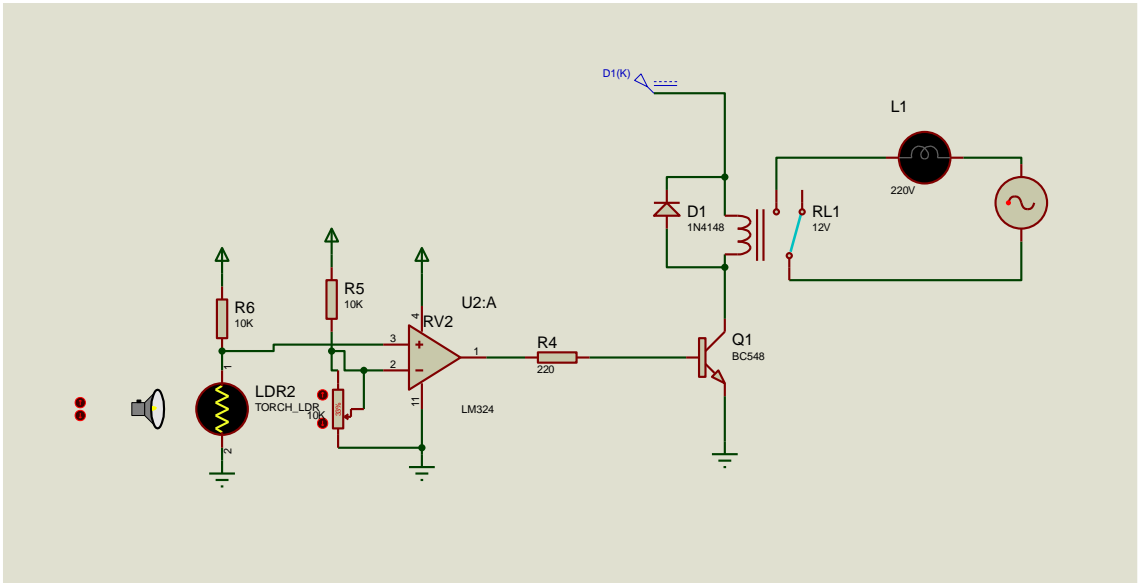
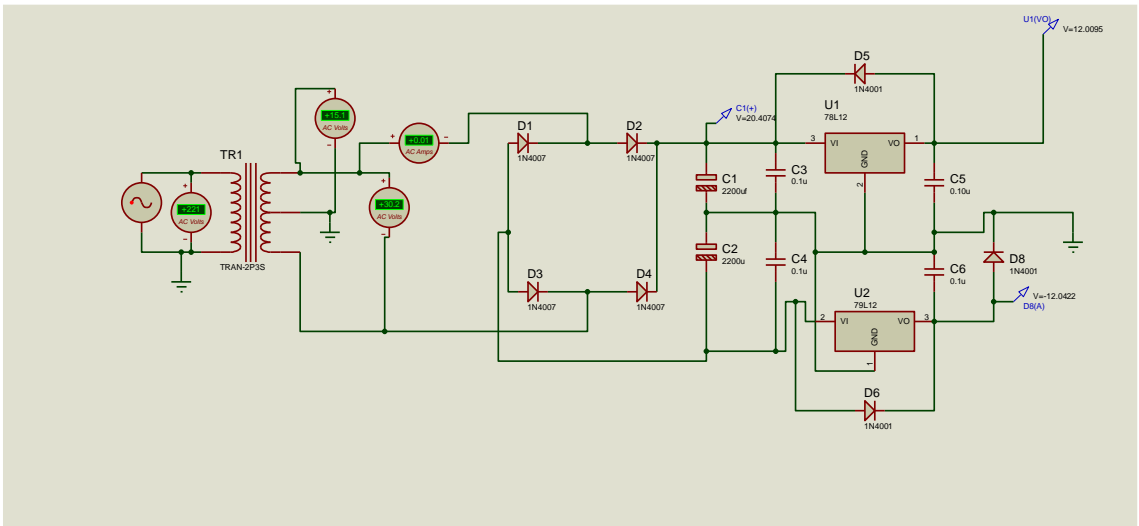


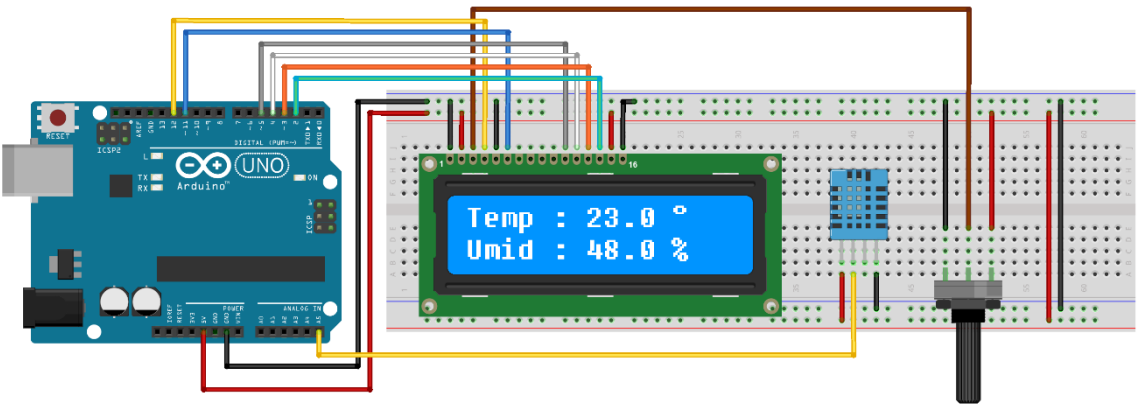
AMPOP COMO COMPARADOR e LDR



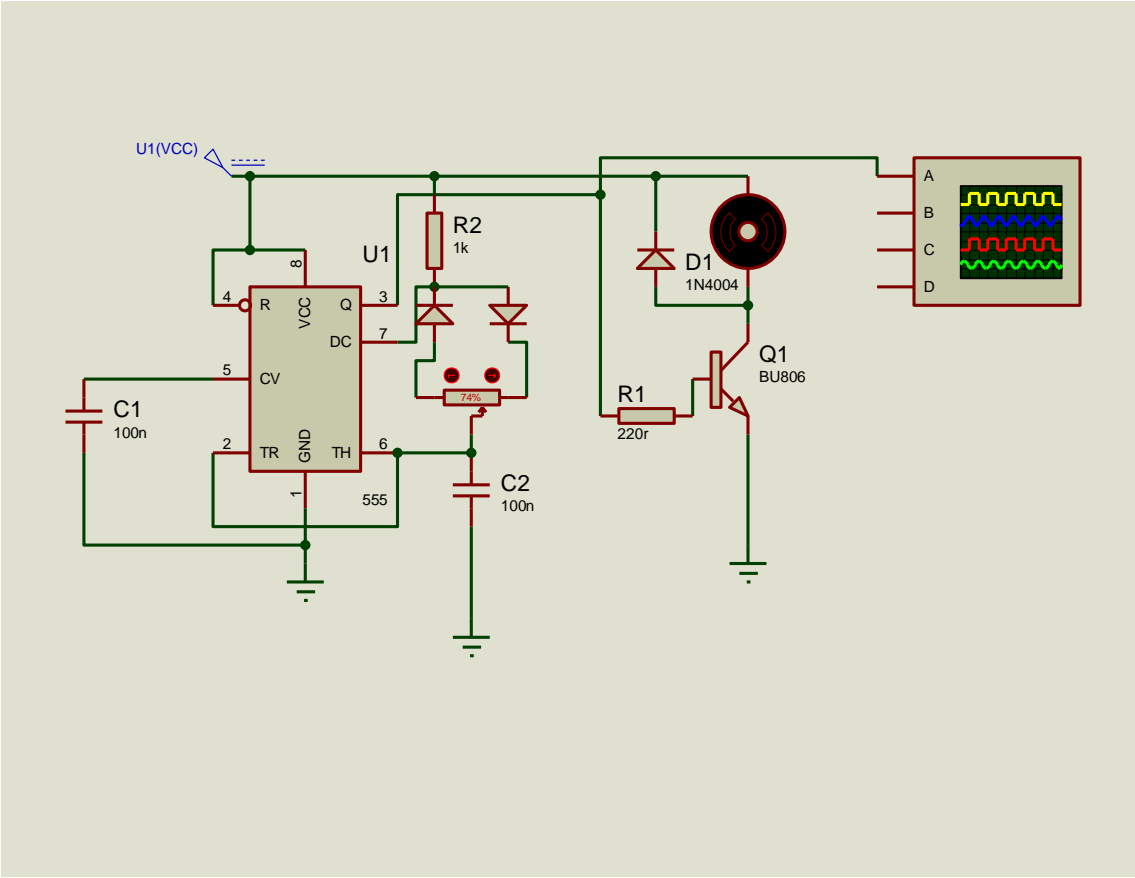
FONTE ALIMENTAÇÃO SIMÉTRICA



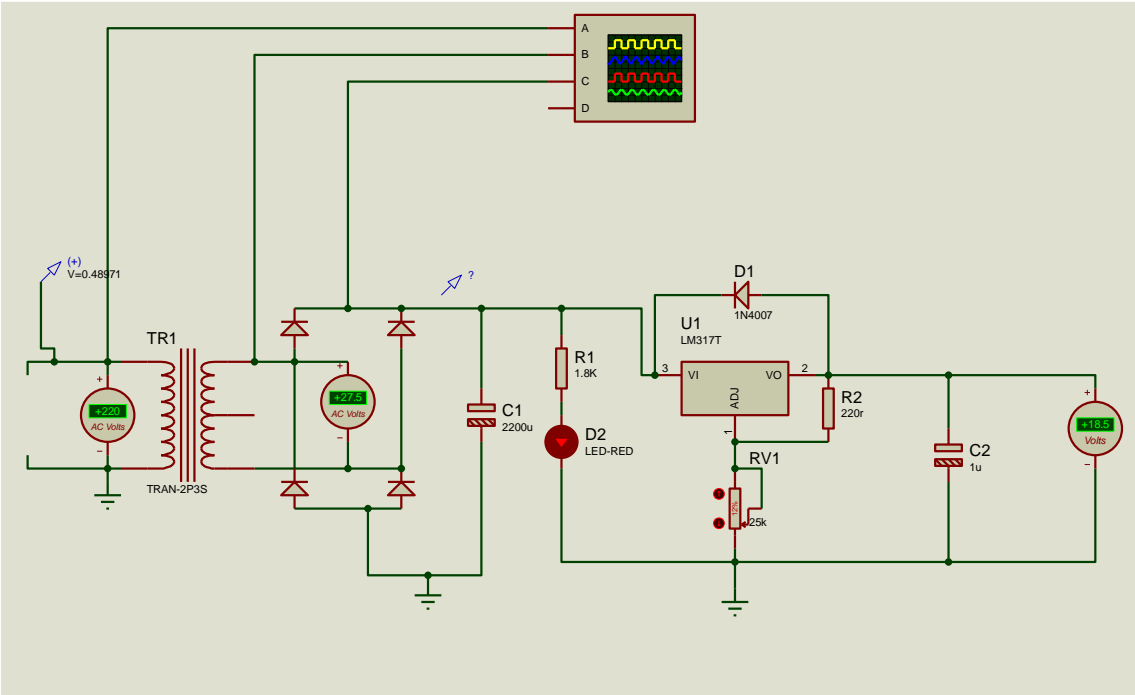
ARDUINO, LCD e DT11



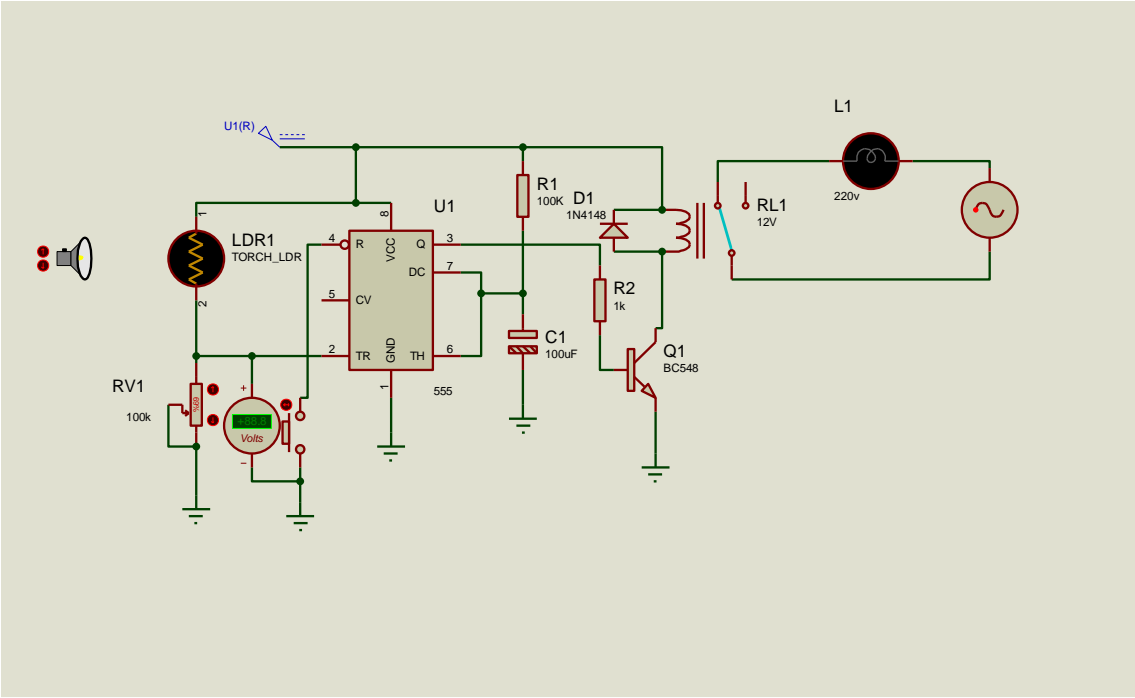
PWM COM 555, TRANSISTOR e MOTOR CC



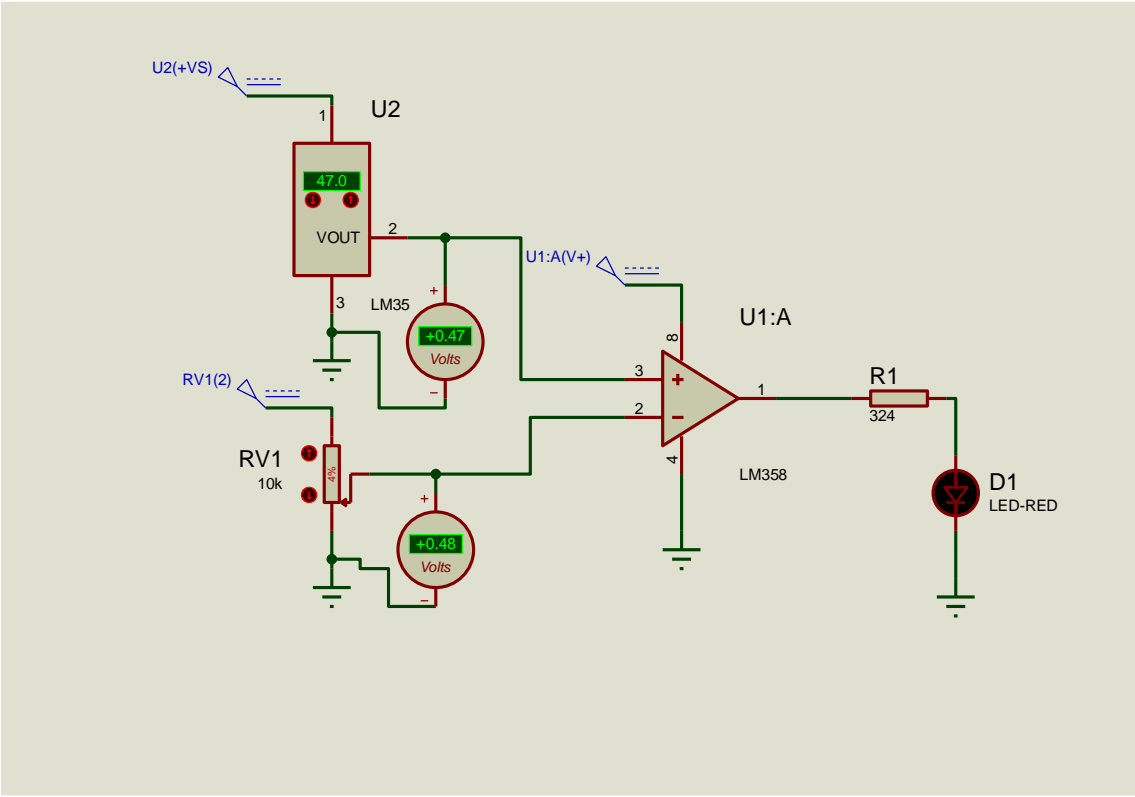
FONTE ALIMENTAÇÃO VARIÁVEL



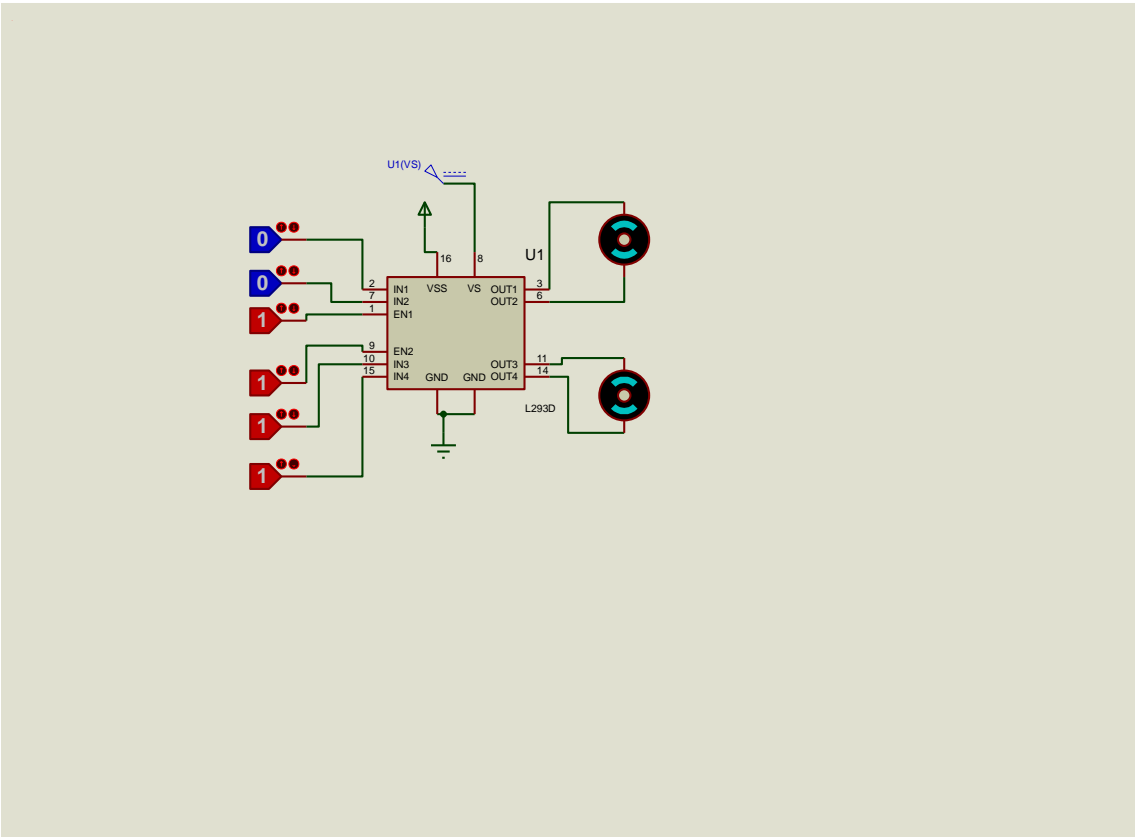
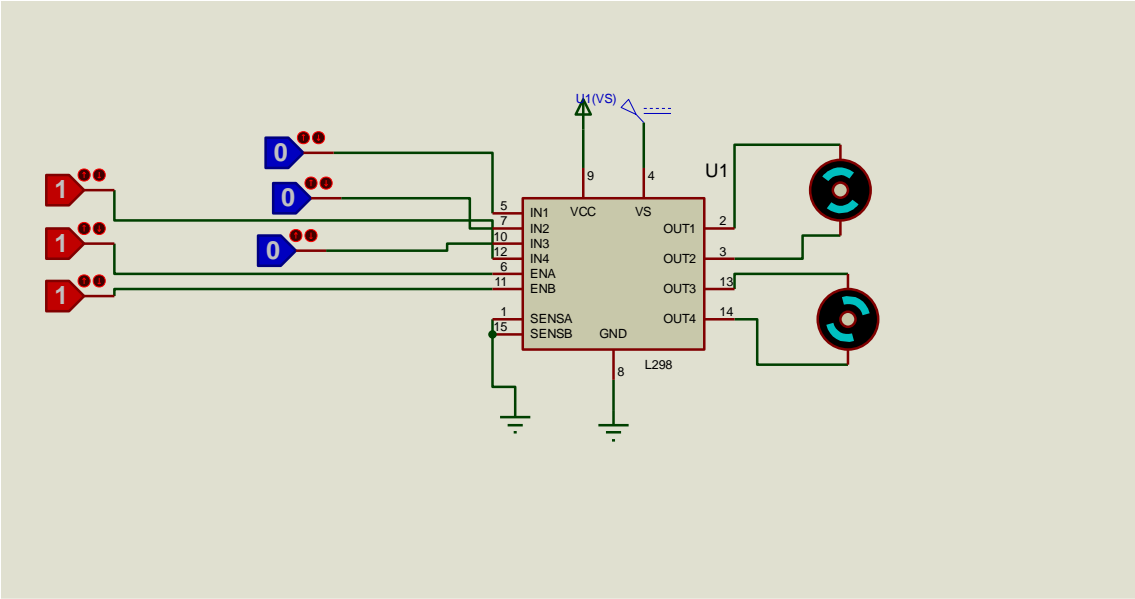
SENSOR FOTOELÉTRICO COM 555, LDR, TRANSISTOR, RELÉ, LÂMPADA 220V



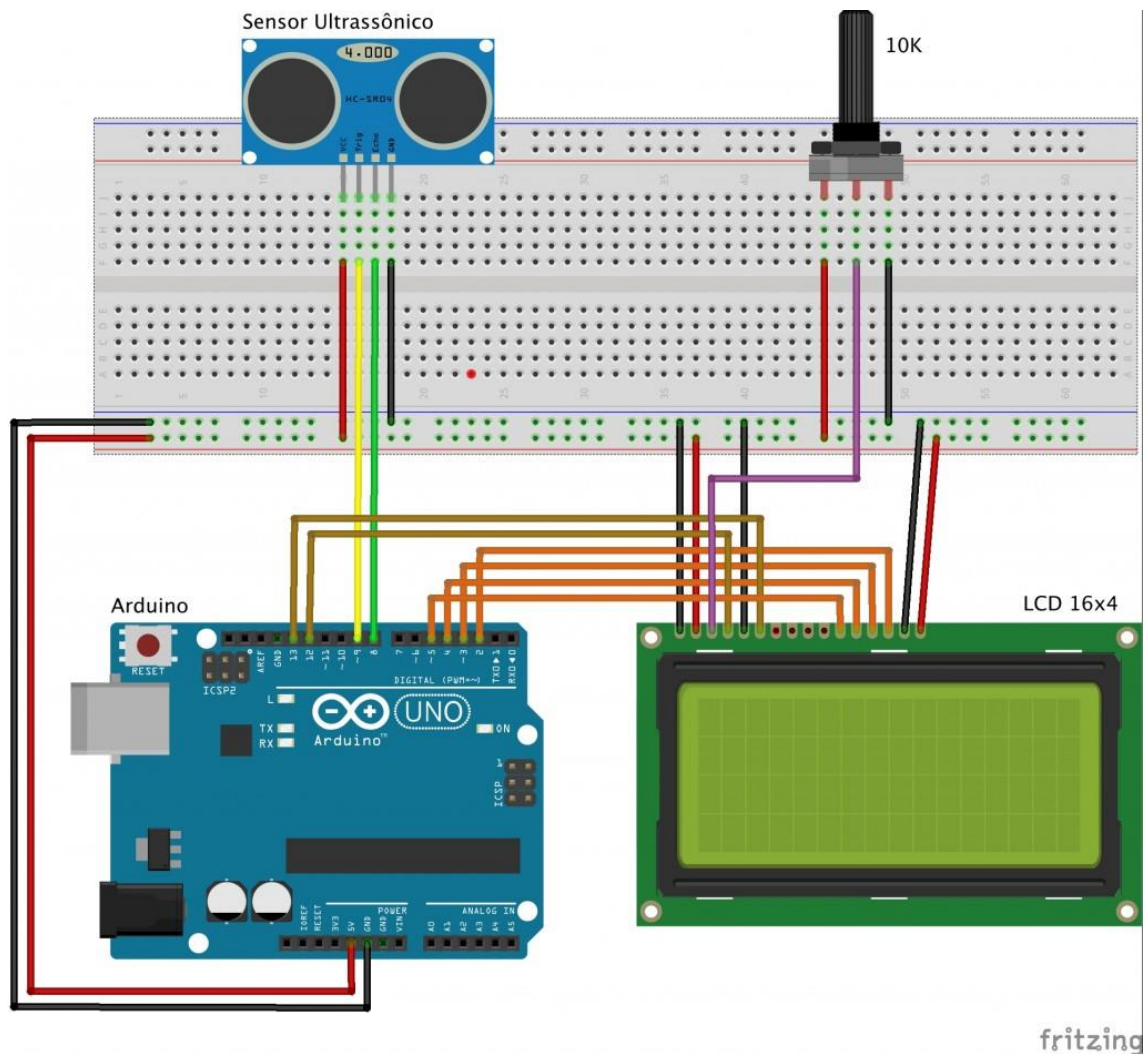
AMPOP COMO COMPARADOR COM SENSOR DE TEMPERATURA



PONTE H COM L293D e L298



SENSOR ULTRASONICO HC SR04/ ARDUINO/LCD



```
// biblioteca para o display LCD
#include <LiquidCrystal.h>

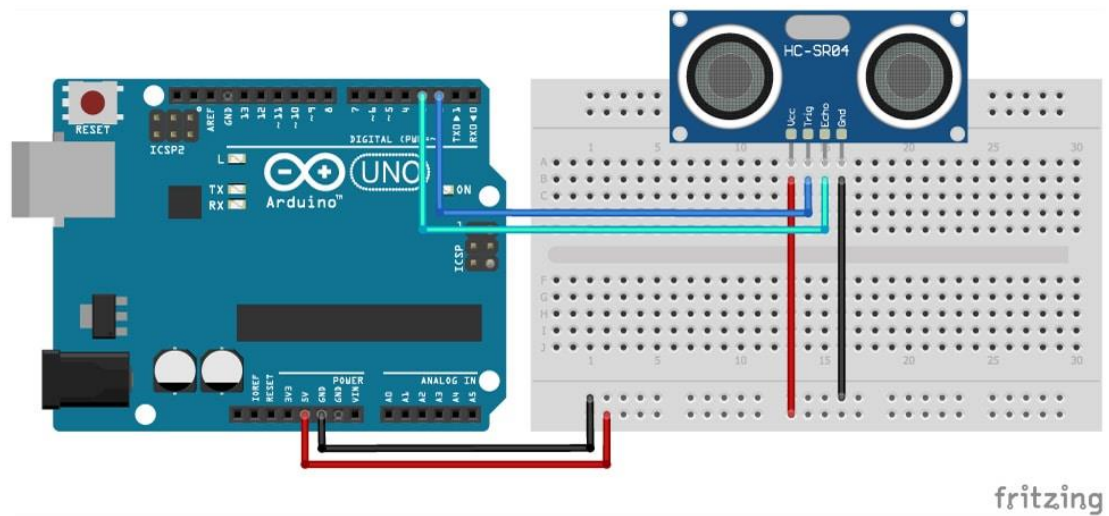
// variáveis
int gatilho = 9; // pino TRIG do sensor ultrassônico
int echo = 8; // pino ECHO do sensor ultrassônico
float tempo; // para armazenar o tempo de ida e volta do
sinal em microsegundos
float distancia_cm; // para armazenar a distância em
centímetros
float distancia_in; // para armazenar a distância em
polegadas

// inicialização do display LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// setup (executado apenas uma vez)
```

```
void setup() {  
  // configura pino GATILHO como saída  
  pinMode(gatilho, OUTPUT);  
  // deixa pino em LOW  
  digitalWrite(gatilho, LOW);  
  delayMicroseconds(10);  
  
  // configura pino ECHO como entrada  
  pinMode(echo, INPUT);  
}  
  
// laço principal (executado indefinidamente)  
void loop() {  
  // disparar pulso ultrassônico  
  digitalWrite(gatilho, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(gatilho, LOW);  
  
  // medir tempo de ida e volta do pulso ultrassônico  
  tempo = pulseIn(echo, HIGH);  
  
  // calcular as distâncias em centímetros e polegadas  
  distancia_cm = tempo / 29.4 / 2;  
  distancia_in = tempo / 74.7 / 2;  
  
  // apresentar resultados no display LCD  
  lcd.begin(20, 4);  
  lcd.setCursor(1, 0);  
  lcd.print("Distancia: ");  
  lcd.print(distancia_cm);  
  lcd.print(" cm");  
  lcd.setCursor(12, 1);  
  lcd.print(distancia_in);  
  lcd.print(" in");  
  lcd.setCursor(3, 3);  
  lcd.print("fazedores.com");  
  
  // aguardar um pouquinho antes de começar tudo de novo  
  delayMicroseconds(200);  
}
```

SENSOR HC SR04 COM ARDUINO



/* Código de exemplo para sensor de distância ultrassônico HC-SR04 com Arduino. Nenhuma biblioteca necessária. */

// Definir Trig e Echo pin:

#define trigPin 2

#define echoPin 3

// Definir variáveis:

long duration;

int distance;

void setup() {

// Definir entradas e saídas:

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Comece a comunicação serial a uma taxa de transmissão de 9600:

Serial.begin(9600);

}

void loop() {

// Limpe o trigPin configurando-o como LOW:

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

// Acione o sensor definindo o trigPin alto por 10 microssegundos:

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Leia o echoPin, pulseIn() retorna a duração (comprimento do pulso) em microssegundos:

duration = pulseIn(echoPin, HIGH);

// Calcule a distância:

distance = duration * 0.034 / 2;

// Imprima a distância no Monitor Serial (Ctrl + Shift + M):

```
Serial.print("Distance = ");  
Serial.print(distance);  
Serial.println(" cm");  
delay(50);  
}
```