

```

#define triac 3 // define o pino de controle do triac
#define LDR A0 // define o pino de entrada do LDR

int tempo_gatilho = 128;

void setup() {
  pinMode(LDR, INPUT); // define o LDR como entrada
  pinMode(triac, OUTPUT); // define o pino de controle do triac como
  saida
}

void loop() {
  int valorLDR = analogRead(LDR); // salva o valor lido do LDR na variavel
  valorLDR
  tempo_gatilho = 128 * valorLDR; // faz a proporcional para calcular o tempo
  de acionamento do triac
  /*
   * Calculo da proporcional:
   * periodo da rede = 1/60Hz = 16.6ms = 16666.6us
   * (16666.6us - 10us (do delay de debounce abaixo)) / 128 = aproximadamente 128
   */
  attachInterrupt(0, zero, FALLING); // gera uma interrupcao toda vez que o
  circuito detectar um zero (inversao de fase)
  delayMicroseconds(1000); // periodo da rede é 16.6ms (1/60Hz), 10 x
  frequencia da rede é 1,66ms. atualizacao do laco maior que 10x a frequencia da
  rede (1ms)
}

void zero() { // funcao que trata a interrupcao e detectado
  uma inversao de fase
  delayMicroseconds(tempo_gatilho); // assim que detecta a inversao espera o tempo
  calculado na proporcional do loop
  digitalWrite(triac, HIGH); // aciona o gatilho do triac
  delayMicroseconds(10); // delay de debounce
  digitalWrite(triac, LOW); // desaciona o gatilho do triac
}

```