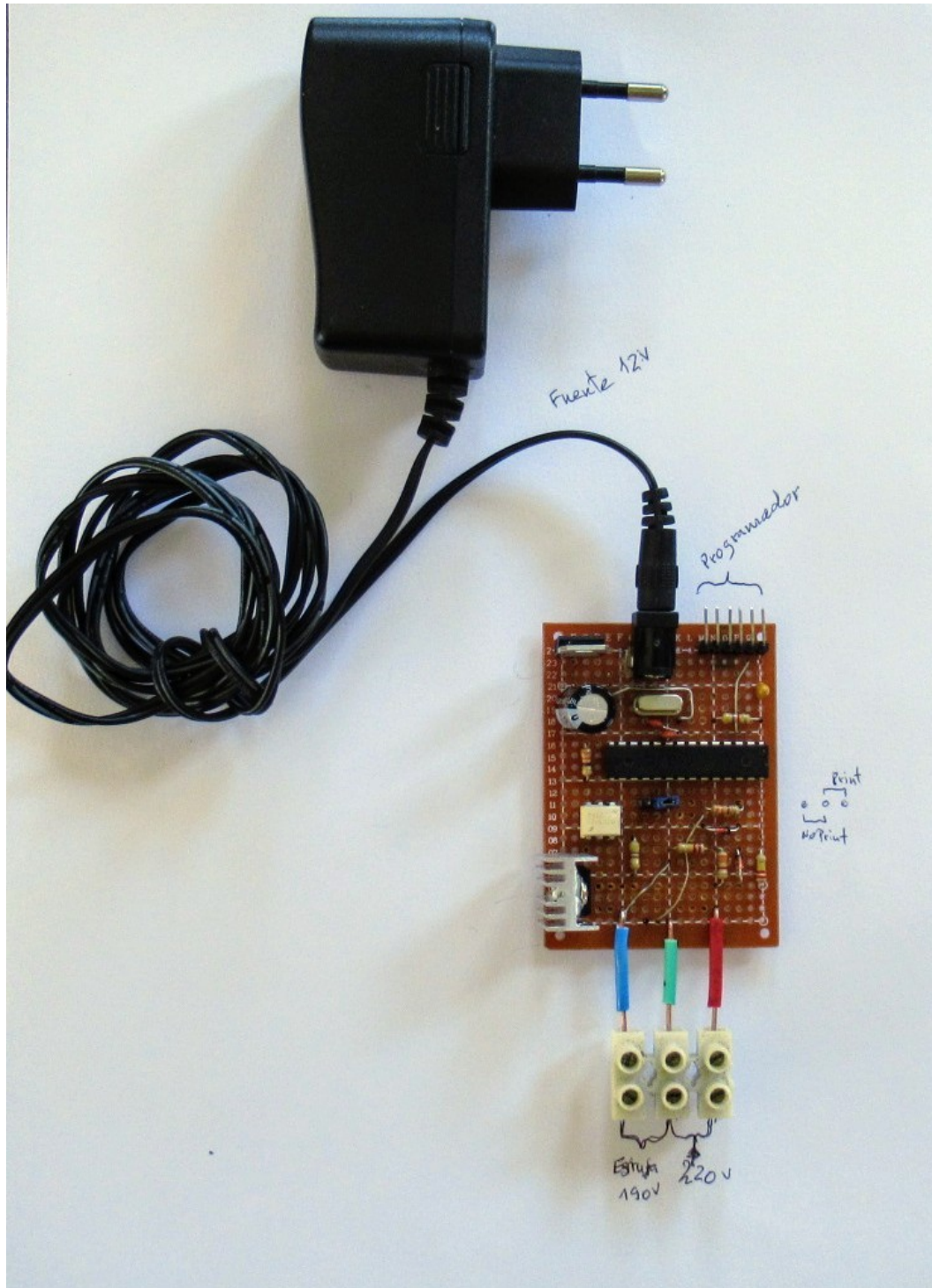


Estabilizador de voltaje

Objetivo

En algunos casos es mas importante la estabilidad junto con un menor voltaje que el nominal. Para esto construimos un control con un triac el que es actuado desfasado, disminuir así el voltaje efectivo.



El triac utilizado es un BTA12-600 que admite una corriente máxima efectiva de 12 amperes. Apresen-ta una caída de voltaje de 1 volt. Con esto para un consumo de 300 watts la corriente es de 1,5 amperes y la energía a disipar en el triac es $1,5 \times 1 = 1,5$ watt. Un pequeño disipador es suficiente.

El programa utilizado es: hay que ajustar Vref (185 volt) y ciclo (16667 microseg)

```
#include <Arduino.h>;
unsigned long T0,Ta,T,t0,t1,t2,Ts,Tciclo,Timp,Tb;
unsigned long Ovm = 2147473647;
float Vmed, Area, Vmeda,Dife,V;
int V0i,Vai,Vi,Vinmax,Vmax,v0i,v1i,v2i,v1,v2,Mciclo;
float Vref = 185;
float Conv = 0.177;
int Tp,k,Vin,Triac, Tpd,n,Inp,Print;
int ciclo = 16667;
int pinTriac=8;
int pinVtriac;
int pinVline;
int pinPrint = 13;
boolean FinPulso;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(pinTriac,OUTPUT);
  pinMode(9,INPUT);
  Mciclo = ciclo / 2;
  Tp=1000;
  Tpd = 1000;
  Triac = 0;
  Timp = millis();
  Serial.println(" parte");
  Vmeda = 0.0;
  Vmed = 0.0;
}

void loop() {
  V0i = 1;
  Vai =1;
  Ta = micros();
  Tb = Ta + Mciclo;
  n=0;
  T0 = Ta;
  Vi=0;
  T= Ta;
  //achar ponto. zero
  while (!(V0i == 0) and (Vai == 0) and (Vi > 0)) {
    V0i = Vai;
    Vai = Vi;
    T0 = Ta;
    Ta = T;
    T=micros();
    Vi = analogRead(A5);
  }
  //achou partida
  Ts = valor_medio(Ta,T); // em algum lugar entre T0 e T
  //Serial.print("achou partida ");
  delayMicroseconds(Tp-100);
  digitalWrite(pinTriac,HIGH);
  Triac = 1;
  // medir volt efetivo

  k = 0;
  t0 = Ts+Tp+Tpd;
  Area = 0.0;
  FinPulso = false;
  Vmax = 0;
  while (!(dif_tempo(micros()),(Ts + Mciclo))) { // aqui alos Tpd micros termina pulso e medimos 1/2 ciclo da voltagem
    t2 = micros();
    v2 = analogRead(A3);
    if (!FinPulso and dif_tempo(t2,t0)) {
      digitalWrite(pinTriac,LOW);
      Triac = 0;
      FinPulso = true;
    }
  }
}
```

```

if (k > 0) { // calcula area
  if ((v1 > v2) and (Vmax == 0)){
    Vmax = v1;
    Vinmax=analogRead(A5);
  }
  Area = Area + (v1+v2)*resta_tempo(t2,t1);
}
t1 = t2;
v1=v2;
k = k + 1;
}
if ((Vmeda > 0.0) or (Vmed == 0.0)) {

  V = Conv*Area/(Mciclo);
  if ( V > 0.9*Vref) {
  // corrigir Tp
  Vmed = V;
  Vmeda = Vmed;
  Dife = ((Vmed+Vmeda)/2.0)-Vref;
  if ( abs(Dife) > 2.2) {
    Tp = Tp + 25*Dife;
    if (Tp < 0 ) Tp = 0;
  }
}
}
while(!dif_tempo(micros(),(Ts + Mciclo + Tp))) {
// espera ate o inicio do 1/2 ciclo negativo
}
digitalWrite(pinTriac,HIGH);
delayMicroseconds(Tpd);
digitalWrite(pinTriac,LOW);
// o tempo que demoa pasa de 13 mseg, mais de 1/2 ciclo, desconectado em um ciclo
Print = digitalRead(pinPrint);
if (dif_tempo(millis() , (Timp + 3000)) and (Print > 0)) {
  Timp = millis();
  Serial.print(Vmed);
  Serial.print("");
  Serial.print(Vref);
  Serial.println("");
}
}

```

```

boolean dif_tempo(unsigned long T2, unsigned long T1) {
unsigned long t1,t2;
t2 = T2;
t1 = T1;
if ((t2 < ciclo) and (t1 > Ovm)) {
  t2 = t2 + Ovm;
  t1 = t1 - Ovm;
} else {
  if ((t1 < ciclo) and (t2 > Ovm)) {
    t2 = t2 - Ovm;
    t1 = t1 + Ovm;
  }
}
if (t2 > t1) {
  return true;
} else {
  return false;
}
}

```

```

unsigned long valor_medio(unsigned long T2, unsigned long T1) {
unsigned long t1,t2;
t2 = T2;
t1 = T1;
if ((t2 < ciclo) and (t1 > Ovm)) {
  t2 = t2 + Ovm;
  t1 = t1 - Ovm;
} else {
  if ((t1 < ciclo) and (t2 > Ovm)) {
    t2 = t2 - Ovm;
    t1 = t1 + Ovm;
  }
}
return ((t1/2) + (t2/2));
}

```

```

float resta_tempo(unsigned long T2, unsigned long T1) {

```

```
unsigned long t1,t2;
t2 = T2;
t1 = T1;
if ((t2 < ciclo) and (t1 > Ovm)) {
    t2 = t2 + Ovm;
    t1 = t1 - Ovm;
} else {
    if ((t1 < ciclo) and (t2 > Ovm)) {
        t2 = t2 - Ovm;
        t1 = t1 + Ovm;
    }
}
return ((1.0*t2) - (1.0*t1));
}
```