

## Modulo Dimmer Digital para Cargas de 220VAC (oleajorge@yahoo.com)

### Descripción

Módulo Dimmer Digital AC versión 1 es ideal para aplicaciones de control de voltaje y potencia AC a través de microcontroladores o placas Arduinos. El Módulo contiene un Triac BT139 de 16A/800VAC, junto con un circuito detector de cruce por cero optoaislado, un circuito de disparo también optoaislado y una red snubber para cargas inductivas.

### Datos Tecnicos:

Alimentación de bajo voltaje : 5 VCC  
Alimentación de alto voltaje : 220VAC  
Potencia : 1KW (se pueden reforzar las pistas para manejar mas potencia)  
Tamaño : 52x42 mm

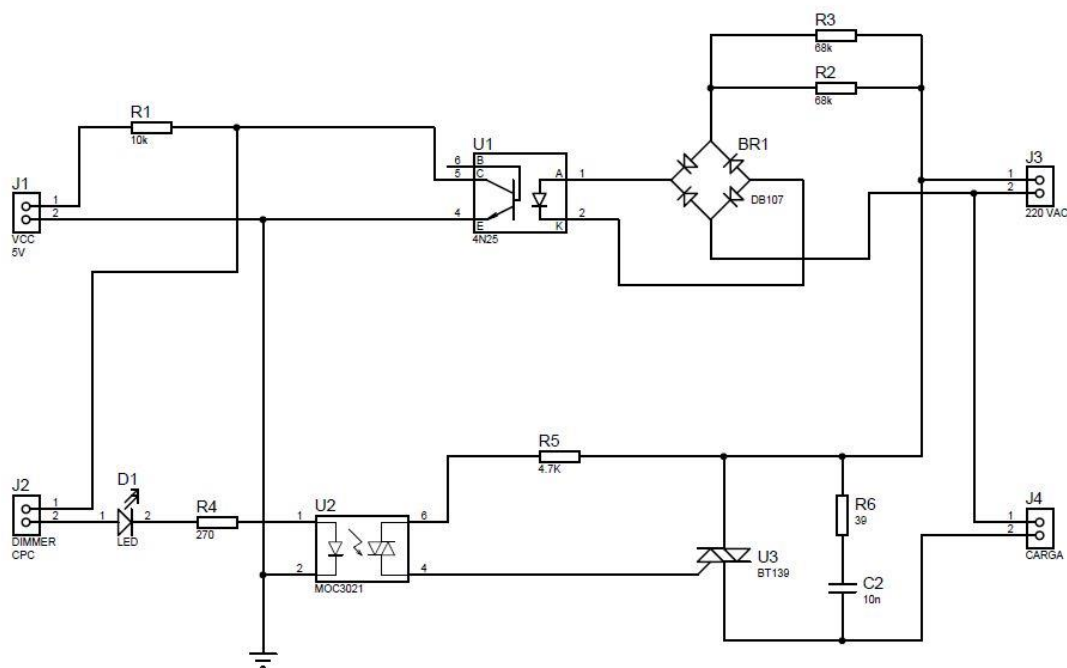
### Area de aplicaciones.

- Control de luminosidad y efectos de lamparas incandescentes
- Control de velocidad de ventiladores
- Control de velocidad de motores AC

### Advertencia:

**Tomar las precauciones de seguridad en atención a que hay un alto riesgo de accidente al operar voltajes altos.**

### Conexiones:



Conecte alimentacion al conector J1-1 (5 VCC)

Conecte alimentacion al conector J1-2 (5 GND)

J2-1 es salida que indica los cruces por cero (CCERO), pin 2 para el Arduino UNO  
J2-2 es entrada para disparar el TRIAC (DISPARO), pin 3 para el Arduino UNO o el que desee.

Conecte el suministro eléctrico al conector azul J3-1 (IN VAC).  
Conecte el suministro eléctrico al conector azul J3-2 (IN VAC).

Conecte la carga a controlar al conector azul J4-1 (CARGA)  
Conecte la carga a controlar al conector azul J4-2 (CARGA)

Sobre J2 hay un LED amarillo que indica cuando hay señal de disparo  
Se adjunta diagrama esquemático  
Hay código fuente para Arduino

### **Código Arduino probado y funcionando:**

```
/*  
AC Light Dimmer v.2 - Inmojo  
AC Voltage dimmer with Zero cross detection  
Author: Charith Fernando http://www.inmojo.com charith@inmojo.com  
License: Released under the Creative Commons Attribution Share-Alike 3.0 License.  
http://creativecommons.org/licenses/by-sa/3.0  
Target: Arduino  
Attach the Zero cross pin of the module to Arduino External Interrupt pin  
Select the correct Interrupt # from the below table
```

Pin	Interrupt #	Arduino Platform
-----	-------------	------------------

2	0	All
3	1	All
18	5	Arduino Mega Only
19	4	Arduino Mega Only
20	3	Arduino Mega Only
21	2	Arduino Mega Only

Please select your utility power frequency from frq variable.

```
*/  
#define fullOn 10  
#define fullOff 127  
#define FQ_50 1 // in case of 50Hz  
//#define FQ_60 0 // in case of 50Hz  
#define VER "2.0"
```

```

int inbyte;
int AC_LOAD = 3; // Output to Opto Triac pin
int dimming = fullOff; // Dimming level (0-128) 0 = ON, 128 = OFF

boolean frq = FQ_50; // change the frequency here.
boolean countUp = false;
boolean simulate = true;

void setup()
{
  pinMode(AC_LOAD, OUTPUT); // Set the AC Load as output
  pinMode(4, OUTPUT);
  attachInterrupt(0, zero_crosss_int, RISING); // Choose the zero cross interrupt # from the table
  above
  Serial.begin(115200);
}

void zero_crosss_int() // function to be fired at the zero crossing to dim the light
{
  // Firing angle calculation
  // 50Hz-> 10ms (1/2 Cycle) → (10000us - 10us) / 128 = 78 (Approx)
  // 60Hz-> 8.33ms (1/2 Cycle) → (8333us - 8.33us) / 128 = 65 (Approx)
  int dimtime = 0;
  float propTime = 0;
  if(frq){
    dimtime = (77*dimming);
    propTime = 10.0;

  }
  else{
    dimtime = (65*dimming);
    propTime = 8.33;
  }
  delayMicroseconds(dimtime); // Off cycle
  digitalWrite(AC_LOAD, HIGH); // triac firing
  delayMicroseconds(propTime); // triac On propogation delay
  digitalWrite(AC_LOAD, LOW); // triac Off

  digitalWrite(4, HIGH); //
  delayMicroseconds(10); // triac On propogation delay

```

```
    digitalWrite(4, LOW);  
}
```

```
void loop()  
{  
    if(simulate){  
        if(countUp)  
            dimming++;  
        else  
            dimming--;  
    }  
}
```

```
if(dimming < fullOn)  
    countUp = true;  
else if (dimming > fullOff)  
    countUp = false;
```

```
delay(20);
```

```
if (Serial.available() > 0)  
    _serial_int();  
}
```

```
void displayMenu() {  
    Serial.println(" ----- InMojo Digital Dimmer v.2 | inmojo.com ----- ");  
    Serial.println("");  
    Serial.println("[m] Menu");  
    Serial.println("[d] Simulate Diming");  
    Serial.println("[1] Turn ON Light");  
    Serial.println("[0] Turn OFF Light");  
    Serial.println("[q] Query Dimming Amount");  
    Serial.println("[v] Version");  
    Serial.println("");  
}
```

```
void _serial_int(){  
    while (Serial.available() > 0) {
```

```

inbyte = Serial.read();

switch (inbyte) {
case 'd':
    attachInterrupt(0, zero_crosss_int, RISING);
    if(simulate)
        simulate = false;
    else
        simulate = true;
    break;
case 'm':
    displayMenu();
    break;
case '1':
    simulate = false;
    detachInterrupt(0);
    digitalWrite(AC_LOAD, HIGH);
    break;
case '0':
    simulate = false;
    detachInterrupt(0);
    digitalWrite(AC_LOAD, LOW);
    break;
case 'q':
    Serial.print("Dimming: ");
    Serial.println(dimming);
    break;
case 'v':
    Serial.println(VER);
    break;
}

}

}

```

