

# Regulación de la Potencia Eléctrica

## Aplicaciones Personales

---

---

### Dimmer para Lampara usando el PIC16F84

---

---

Autor: Carlos A. Narváez V. Ingeniero Electricista Universidad de Oriente Email: cnarvaez@bolivar.udo.edu.ve
---

#### Propósito

Existe gran cantidad de aplicaciones donde se requiere la regulación de la corriente alterna, entre ellas, el control de velocidad de motores, la soldadura eléctrica y la cantidad de iluminación. Esto se puede lograr con el uso de autotransformadores o introduciendo resistencias variables. Ninguno de estos dos métodos resultan aconsejables, el primero resulta muy caro y el segundo muy ineficiente. El desarrollo de los SCR y los Triac han hecho del control de potencia eléctrica un proceso relativamente sencillo y barato.

El presente proyecto tiene como proposito mostrar la regulación de potencia eléctrica utilizando el PIC16F84 y agregando como hardware adicional un optoacoplador que aísla a este del circuito de potencia constituido por un Triac y una Lampara de 120VAC 100W. La técnica se basa en que la potencia de salida puede variarse regulando la fase de conducción del Triac tanto en el semiciclo positivo como en el semiciclo negativo de la onda sinusoidal.

#### Teoría de Operación

El TRIAC es un interruptor de silicio de tres terminales que se puede disparar (hacer conducir) con impulsos positivos y negativos de la puerta cuando los potenciales del anodo son positivos o negativos respectivamente, es decir pueden conducir en las dos semiondas de la corriente alterna.

Utilizando el optoacoplador K3020P, este tiene un triac el cual al ser disparado por la luz del diodo autocontenido, produce el disparo del triac de potencia BT136 tanto en el semiciclo positivo como en el negativo. Haciendo variar la fase del disparo podemos regular la potencia en la carga.

El circuito contiene un transformador de corriente alterna cuya salida es conectada al pin RA0 del PIC16F84 a fin de sensar el cruce por cero de la onda sinusoidal y poder sincronizar la operación del software. El pin RA4 es utilizado como salida de los pulsos que disparan el optoacoplador. Dos pulsadores conectados a los pines RB0 y RB1 son utilizados para elevar y bajar respectivamente la potencia suministrada a la carga.

## Regulación de la Potencia Eléctrica Aplicaciones Personales

---

La fig. 1 muestra la tensión de entrada, y la tensión aplicada a la carga. Durante el ángulo de retraso el triac no conduce, por lo que la carga no recibe tensión, luego se dispara (conduce) y la parte que falta para completar el semiciclo positivo es aplicada a la carga. Esto se repite para el semiciclo negativo. El control de los tiempos de disparo del triac nos permite regular la tensión aplicada a la carga.

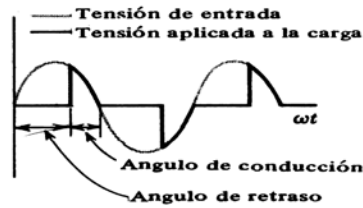


Fig 1 Tensión de entrada y aplicada a la carga

La fig. 2 muestra la relación del ángulo de conducción versus el voltaje aplicado a la carga para un triac típico. Las curvas muestran la tensión pico a pico, la tensión efectiva o RMS y el promedio de ambas.

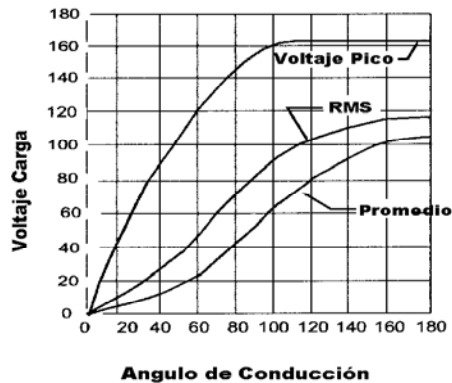


Fig. 2 Voltaje en la carga versus Ángulo de Conducción del Triac

## Regulación de la Potencia Eléctrica Aplicaciones Personales

### Hardware

El hardware diseñado para este proyecto se muestra en la figura 3. Este circuito puede manejar cargas de 120VAC de hasta 100watt. El Pin MCLR el cual es activo bajo, se configura como entrada y es usado como Reset del microcontrolador. La frecuencia de trabajo es de 4Mhz y todo el circuito es alimentado con +5 voltios. Los pulsadores son utilizados para incrementar y disminuir la tensión aplicada a la carga: una lampara incandescente.

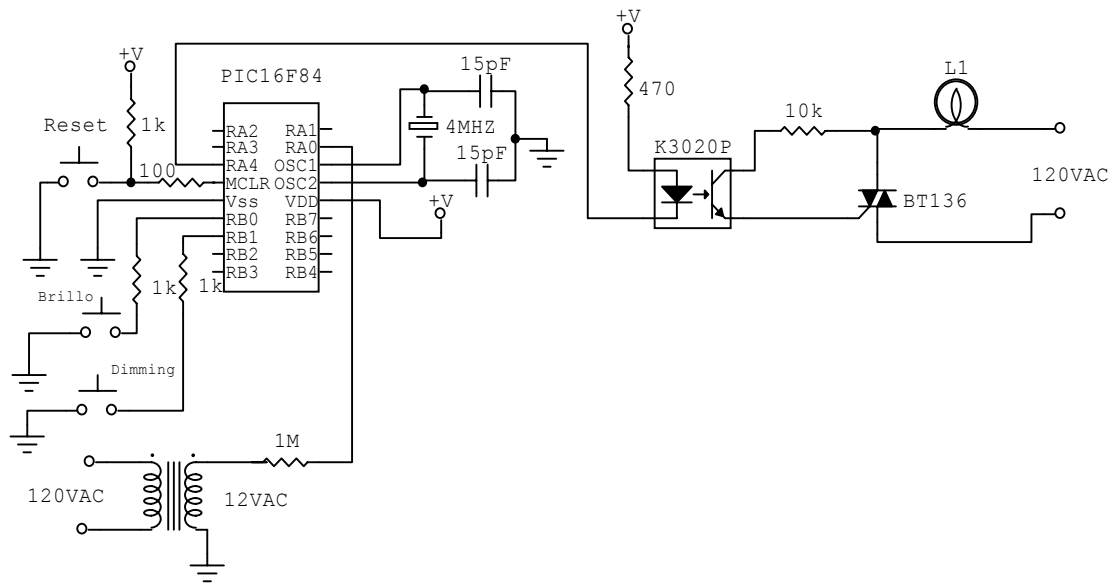


Fig. 3 Controlador de potencia electrico basado en el PIC16F84

# Regulación de la Potencia Eléctrica Aplicaciones Personales

---

## Software

```
/*
 * Programa PicDimmer - Controlando la Potencia Eléctrica con PIC
 * Rutinas originales: Scott Fink Microchip 1997
 * Adaptación PIC16F84: Carlos Narvaez, 2002
 * Compilado: CC5X Free V3.1
 *
 * RB0 = Pulsador Dim
 * RB1 = Pulsador Brillo
 * RA0 = Sensor de cruce por cero
 * RA4 = Salida al optoacoplador
 */

#include <16f84.h>
#pragma bit Brtbut @ PORTB.0
#pragma bit Dimbut @ PORTB.1
#pragma bit LineInput @ PORTA.0
#pragma bit Salida @ PORTA.4

void Buttoncheck(void); //verificar los pulsadores
unsigned int PercentOn, Maxdim; //variables globales
unsigned int TestCheck, Outcount, TestCount;
unsigned int DelayCnt;
unsigned int LastBoth, FirstPass;
unsigned int Count;
unsigned int Maxbrt, NotInTest;

void main() {
    Maxbrt = 0xf0; //brillo máximo
    NotInTest = 3;
    PercentOn = 0xd0; //periodo encendido
    Maxdim = 0x70; //valor máximo Dim
    TestCheck = 0; //
    Outcount = 0; //contador salida del modo test
    TestCount = 0; //contador modo test
    DelayCnt = NotInTest; //contador retardo
    LastBoth = 0; //bandera ambos pulsadores
    //presionados la ultima vez

    FirstPass = 1; //
    Count = 0; //contador general

    for(Count = 0; Count < 60; Count++) //estabilización tensión
    {
        while(LineInput == 1);
        while(LineInput == 0);
        clrwdt();
    }
    W = 0x5; //1:64 prescale, Pull-up activo
    OPTION = W;

    W = 0x11; //RA0, RA4 Entradas
    TRISA = W;
    PORTA = 0x10; //RA4 Latch high

    W = 0x3; //RB0, RB1 Entradas
    TRISB = W;

    while(LineInput == 1) //Sincronización fase linea
        clrwdt();
    TMR0 = PercentOn;
    while(TMR0 >= 3 && LineInput == 0) //retardo entrada punto apropiado
        clrwdt();
}
```

## Regulación de la Potencia Eléctrica Aplicaciones Personales

---

```
while(1)
{
    Count = 0;
    while(Count < DelayCnt)
    {
        Count += 1;
        if(LineInput == 1)
        {
            Maxdim +=5;
            while(LineInput == 1);
            while(LineInput == 0)
                clrwdt();
        }
        else
        {
            while(LineInput == 0)
                clrwdt();
            Maxdim = PercentOn - TMR0 + 2;
        }
        if(FirstPass == 1)
        {
            FirstPass = 0;
            PercentOn = Maxdim;
        }
        if(PercentOn < Maxdim)
            PercentOn = Maxdim;
        TMR0 = PercentOn;
        while(TMR0 >= 3 && LineInput == 1)
            clrwdt();
        PORTA = 0x00;
    }
    W = 0x01;
    TRISA = W;
    #asm
        nop
        nop
        nop
        nop
        nop
        nop
        nop
    #endasm
    W = 0x11;
    TRISA = W;
    clrwdt();
    if(LineInput == 0)
    {
        Maxdim +=5;
        while(LineInput == 0);
        while(LineInput == 1)
            clrwdt();
    }
    else
    {
        while(LineInput == 1)
            clrwdt();
        Maxdim = PercentOn - TMR0 + 2;
    }
}
```

## Regulación de la Potencia Eléctrica Aplicaciones Personales

---

```
        if(PercentOn < Maxdim)                //corrija brillo
            PercentOn = Maxdim;
        TMR0 = PercentOn;                    //Retardo
        while(TMR0 >= 3 && LineInput == 0)
            clrwdt();
        PORTA = 0x00;
W = 0x01;
TRISA = W;                                //dispere triac
#asm

        nop
        nop                                //retardo señal disparo
        nop
        nop
        nop
        nop
        nop
        nop
#endasm
W = 0x11;
TRISA = W;                                //Fin señal disparo
        clrwdt();
    }
    Buttoncheck();                          //verifica pulsadores
}
}
/*****
* Subrutina ButtonCheck
*
* Esta rutina verifica si han sido presionados los pulsadores de Brillo o Dim
* e incrementa o decrementa la variable PercentOn.
*
* Si ambos pulsadores han sido presionados y la lampara no esta apagada, esta
* se apaga. Si esta apagada entonces esta es encendida a máximo brillo.
*
* Modo Test: Si ambos pulsadores son presionados al mismo tiempo y luego se
* libera el pulsador Dim, presionandolo nuevamente, se entra en el modo test.
* Si el pulsador Dim es presionado estando en el modo test, entonces el
* programa sale al modo normal, con la lampara apagada. El modo test eleva
* el brillo hasta el máximo, luego disminuye el brillo hasta full dim y luego
* palpadea a maximo brillo dos veces, luego se repite.
*****/
void Buttoncheck()
{
    nop();
    if(TestCheck == 3)                      //Verifica bandera de modo test
    {
        DelayCnt = 2;                       //reset contador de retardo
        if(Brtbut && !Dimbut)                //Si pulsador Dim esta presionado
        {                                    //salga del modo test
            TestCheck = 0;                   //Bandera modo test 0
            DelayCnt = 5;
            return;
        }
        if(TestCount == 0)                  //Eleve brillo hasta Maxbrt
        {
```

## Regulación de la Potencia Eléctrica Aplicaciones Personales

---

```
    if(++PercentOn > Maxbrt)                //verifique brillo máximo
    {
        PercentOn = Maxbrt;
        TestCount += 1;
        return;
    }
    else
        return;
}

if(TestCount == 1)                          //disminuya brillo hasta Maxdim
{
    PercentOn -= 1;
    if(PercentOn <= Maxdim)                 //verifique máximo Dim
    {
        PercentOn = Maxbrt;
        TestCount += 1;
        return;
    }
    else
        return;
}

while(TestCount < 5)                         //retardo
    TestCount += 1;
    return;

while(TestCount < 10)                       //Apagar por un corto periodo
{
    TestCount += 1;
    PercentOn = Maxdim;
    return;
}

while(TestCount < 15)                       //Encender por un corto tiempo
{
    TestCount +=1;
    PercentOn = Maxbrt;
    return;
}

while(TestCount < 20)                       //Apagar por un corto tiempo
{
    TestCount += 1;
    PercentOn = Maxdim;
    return;
}

while(TestCount < 25)                       //Encender por un corto tiempo
{
    TestCount += 1;
    PercentOn = Maxbrt;
    return;
}
```

## Regulación de la Potencia Eléctrica Aplicaciones Personales

---

```
while(TestCount < 30) //Apagar por un corto tiempo
{
    TestCount += 1;
    PercentOn = Maxdim;
    return;
}

PercentOn = Maxdim;
TestCount = 0; //Reset comienzo secuencia de test
if(++Outcount == 255) //Ejecute modo test 255 ciclos
{
    TestCheck = 0; //Bandera de modo test 0
    DelayCnt = NotInTest;
    Outcount = 0;
}
return;
}
if(TestCheck) //
{
    if(++Outcount == 0x60)
    {
        DelayCnt = NotInTest;
        Outcount = 0;
        TestCheck = 0;
    }
}
if(!TestCheck && !Brtbody && !Dimtbody) //ambos pulsador presionados
    TestCheck = 1;
if(TestCheck == 1 && !Brtbody && Dimtbody) //pulsador brillo presionado solo
    TestCheck = 2;
if(TestCheck == 2 && !Brtbody && !Dimtbody) //ambos pulsadores presionados
{
    TestCheck = 3; //entra en modo test
    TestCount = 0;
    PercentOn = Maxdim;
    Outcount = 0;
}
if(!Dimtbody && !Brtbody) //si ambos estan presionados
{
    if(LastBoth == 0)
    {
        LastBoth = 1;
        if(PercentOn == Maxdim) //si apagado total
            PercentOn = Maxbrtbody; //entonces, encendido total
        else
            PercentOn = Maxdim; //caso contrario, apaguelo
    }
}
else
    LastBoth = 0;
if(!Brtbody && Dimtbody) //incrementa brillo
    PercentOn++;
if(Brbtbody && !Dimtbody) //disminuye brillo
    PercentOn--;
if(PercentOn > Maxbrtbody)
    PercentOn = Maxbrtbody;
if(PercentOn < Maxdim)
    PercentOn = Maxdim;
}
```