

Como programar GRAFCET con Arduino. Arranque estrella triángulo con Arduino implementado según GRAFCET

Antonio Blanco Solsona



Revista Digital de ACTA
2020

Publicación patrocinada por



ACTA representa en CEDRO los intereses de los autores científico-técnicos y académicos. Ser socio de ACTA es gratuito.

Solicite su adhesión en acta@acta.es

Como programar GRAFCET con Arduino. Arranque estrella triángulo con Arduino implementado según GRAFCET

© 2020, **Antonio Blanco Solsona**

© 2020,  ACTA

Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley.

Se autorizan los enlaces a este artículo.

ACTA no se hace responsable de las opiniones personales reflejadas en este artículo.

INTRODUCCIÓN

Antes de comenzar se va a describir Arduino y GRAFCET para que se comprenda mejor cómo implementar la automatización esquematizada en Grafcet. La implementación se realiza sobre la placa de desarrollo empleando el lenguaje de programación C.

Arduino es una placa hardware de libre distribución y utilización. La programación se realiza en C a partir del entorno de programación del propio Arduino. Se programa a partir de una conexión USB. El programa se transfiere ya compilado.

En primer lugar, es necesario configurar la comunicación entre Arduino y el ordenador. Deberá comunicarse configurando la misma velocidad y paridad en la comunicación serie.

La placa de Arduino se muestra en la Figura 1

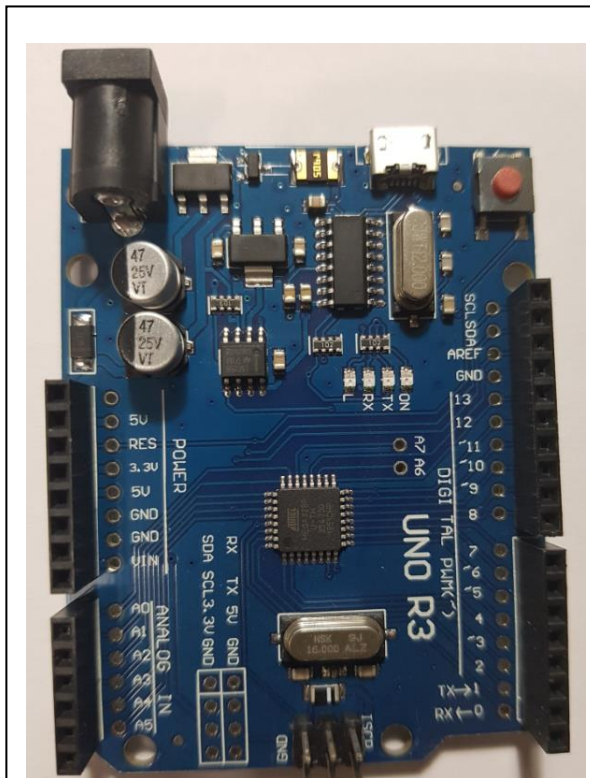


Figura 1. Arduino

En la parte superior izquierda se tiene la alimentación. En la parte superior derecha la conexión USB al ordenador.

En el lateral izquierdo se encuentran los terminales de 5V, 3,3 V, masa y reset. Debajo las 6 entradas analógicas.

En el lateral derecho se encuentran los dos terminales de Comunicaciones, y las entradas o salidas digitales, según la configuración por la que se opte.

Hay que destacar que las entradas deben de adaptarse para conseguir la compatibilidad electromagnética e impedir falsos impulsos por accionamientos de motores o cualquier otra fuente de ruido eléctrico.

El entorno de programación es el indicado en la Figura 2

En herramientas se selecciona el tipo de placa, y se configure la comunicación del puerto. A partir de ese momento ya se puede programar la placa en C.

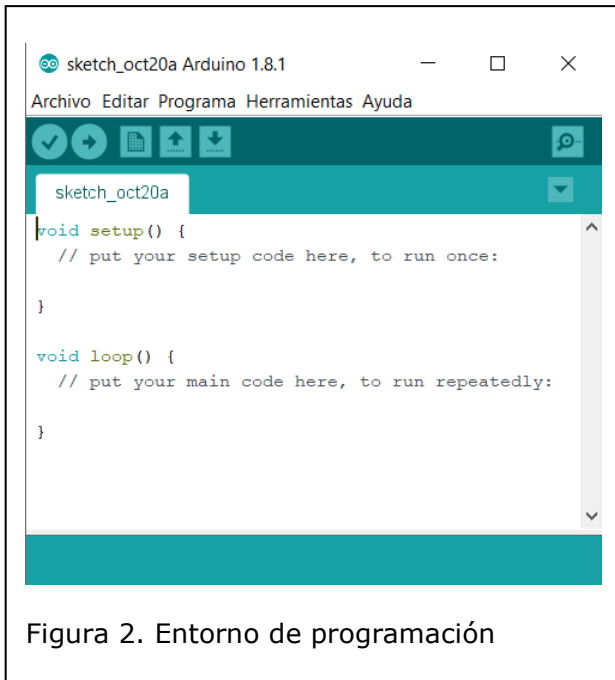


Figura 2. Entorno de programación

Un ejemplo sencillo de salida intermitente es el indicado en la Figura 3.

En el ejemplo de la Figura 3 se configura el terminal 13 como salida. En el bucle principal del programa se hace que dicha salida alterne cada segundo entre un valor alto y bajo. La unidad de tiempo está dada en milisegundos. El bucle se repite indefinidamente.



Figura 3. Ejemplo de salida intermitente

Respecto al GRAFCET, se puede resumir como un diagrama de flujo que representa toda la automatización que se quiere implementar. El GRAFCET es un método gráfico utilizado para representar los automatismos secuenciales. Describe gráficamente la evolución del mismo. El GRAFCET representa la sucesión de estados en una automatización. Cada etapa tiene asociada una acción determinada.

Entre cada etapa se tiene una transición. Las transiciones indican las condiciones para ir a la etapa siguiente. Una de las condiciones es que la etapa anterior esté activa.

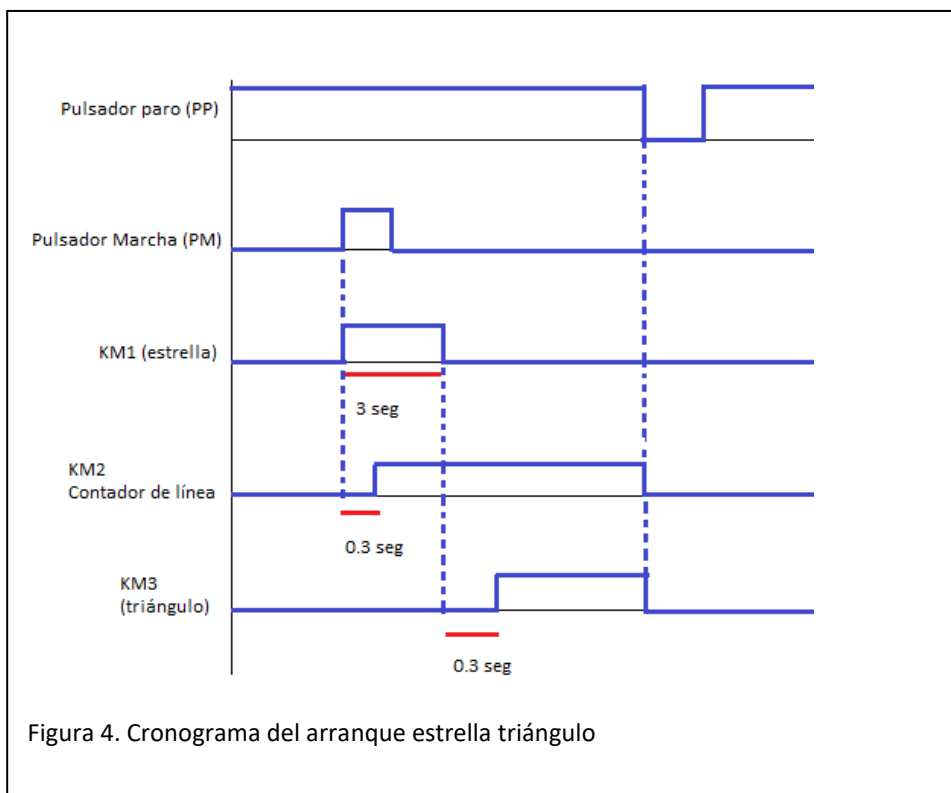
Cuando se pasa al estado siguiente se desactivan los estados anteriores.

Las acciones se representan por un rectángulo a la derecha de la etapa. Corresponden a la parte de potencia o temporizaciones en la automatización de un proceso industrial.

CRONOGRAMA DEL ARRANQUE ESTRELLA TRIÁNGULO

Se va a describir el arranque estrella triángulo de un motor trifásico. El arranque en estrella durará 3 segundos, a los 0,3 segundos de apagarse el arranque en estrella se tiene que activar la conexión en triángulo que controlará el motor hasta que se desconecte mediante el pulsador de paro. En la práctica el tiempo de arranque en estrella se puede determinar experimentalmente conectando el motor en estrella y midiendo con una pinza amperimétrica el tiempo que tarda la corriente en estabilizarse. Este tiempo será el que se programará. En este caso se ha asumido que es de 3 segundos.

El cronograma del funcionamiento es el indicado en la Figura 4.



Este cronograma es el que se utilizará para describir el Grafcet que dará lugar a la implementación en C para su compilación en Arduino.

EL GRAFCET

Antes de diseñar el GRAFCET, se establecen las entradas y salidas de la automatización.

Las entradas se asignan a las entradas digitales 2 y 3 de Arduino UNO. El pulsador de marcha a la entrada digital 2 y el pulsador de paro a la entrada digital 3 según se muestra en la Tabla 1.

Pulsador de marcha	PM	Entrada digital 2
Pulsador de paro	PP	Entrada digital 3

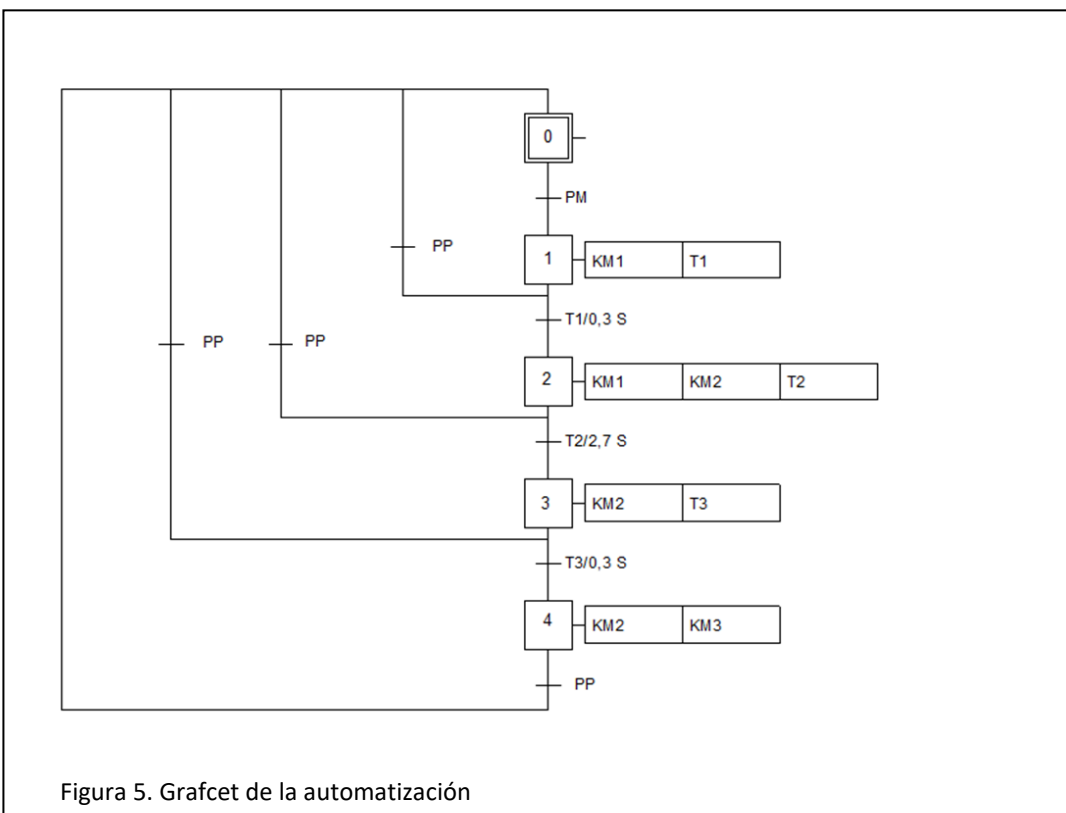
Tabla 1. Entradas digitales

Las salidas se asignan a las salidas digitales 4, 5 y 6 de Arduino UNO. KM1 a la salida digital 4, KM2 a la salida digital 5 y KM3 a la salida digital 6 según se muestra en la Tabla 2.

KM1	Contactador estrella	Salida digital 4
KM2	Contactador línea	Salida digital 5
KM3	Contactador triángulo	Salida digital 6

Tabla 2. Salidas digitales

El GRAFCET que implementa el cronograma anterior es el indicado en la Figura 5.



A los estados se le asignan las variables indicadas en la Tabla 3.

Estados	Variables
0	x0
1	x1
2	x2
3	x3
4	x4

Tabla 3. Estados

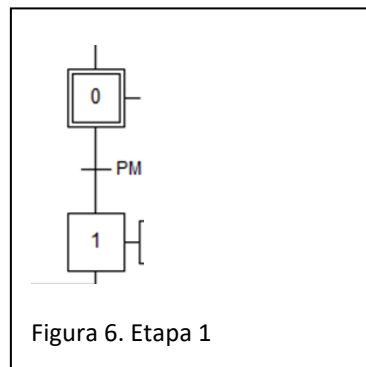
Y se comienza a programar implementando los estados para luego finalmente unir todo el código junto con la configuración correspondiente.

Empezamos por la etapa 1

//etapa1

```

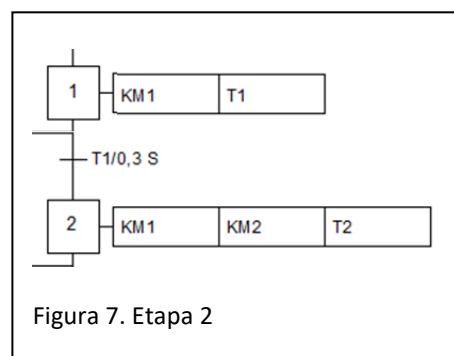
if(x0==1 and pm==1)
{
    x1=1;
    x0=0;
}
    
```



//etapa2

```

if(x1==1 and T1==1)
{
    x2=1;
    x1=0;
    T1=0;
}
    
```



}

//etapa3

if(x2==1 and T2==1)

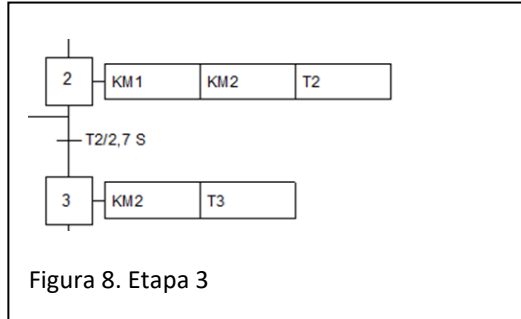
{

x3=1;

x2=0;

T2=0;

}



//etapa4

if(x3==1 and T3==1)

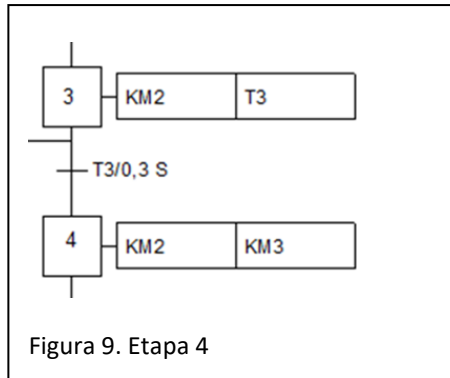
{

x4=1;

x3=0;

T3=0;

}



//etapa0

if((x1==1 or x2==1 or x3==1 or x4==1) and pp==1)

{

x0=1;

x1=0;

x2=0;

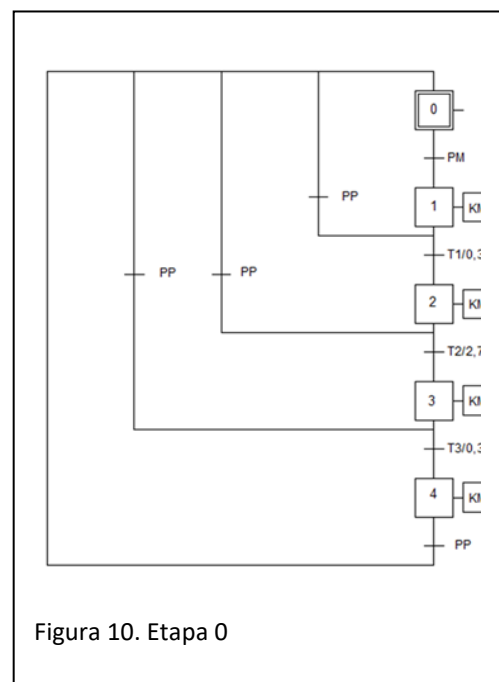
x3=0;

x4=0;

T1=0;

T2=0;

T3=0;




```
}
```

```
//etapa de inicio
```

```
if (x0=0 and x1=0 and x2=0 and x3=0 and x4 = 0)
```

```
{
```

```
  x0=1;
```

```
}
```

```
//etapa de potencia KM1 y T1
```

```
if(x1==1)
```

```
{
```

```
  KM1=1;
```

```
  c=c+1;
```

```
  If(c==300)
```

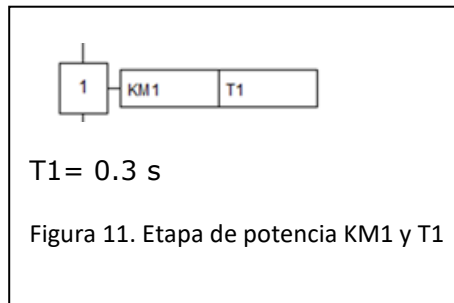
```
    {
```

```
      T1=1;
```

```
      c= 0;
```

```
    }
```

```
}
```



```
//etapa de potencia KM1, KM2 y T1
```

```
if(x2==1)
```

```
{
```

```
  KM1=1;
```

```
  KM2=1;
```

```
  c=c+1;
```

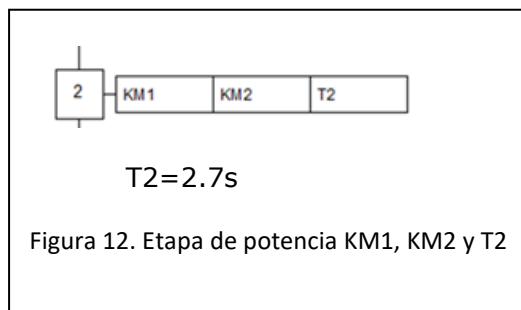
```
  If(c==2700)
```

```
    {
```

```
      T2=1;
```

```
      c= 0;
```

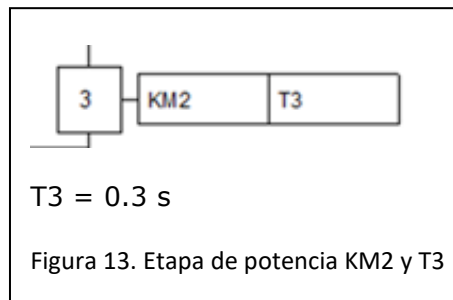
```
    }
```



```
}
```

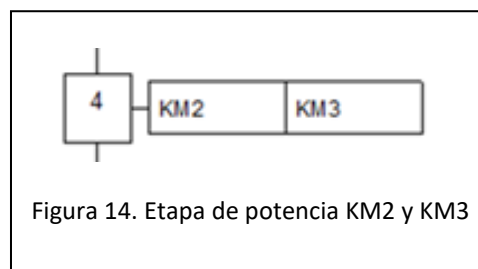
```
//etapa de potencia KM2 y T3
```

```
if(x3==1)  
{  
  KM2=1;  
  c=c+1;  
  If(c==300)  
  {  
    T3=1;  
    c= 0;  
  }  
}
```



```
//etapa de potencia KM2 y KM3
```

```
if(x4==1)  
{  
  KM2=1;  
  KM3=1;  
}
```



```
//base de tiempos en milisegundos
```

```
delay(1);
```

PROGRAMA COMPLETO

Se realiza la programación en C, se compila y se transfiere a Arduino.

```
// Estrella triangulo
```

```
int x0 = 0; // Etapa 0 inicialmente está encendida  
int x1 = 0; // Etapa 1 inicialmente está apagada  
int x2 = 0; // Etapa 2 inicialmente está apagada
```

```

int x3 = 0; // Etapa 3 inicialmente está apagada
int x4 = 0; // Etapa 4 inicialmente está apagada
int T1 = 0;
int T2 = 0;
int T3 = 0;
int pm = 0;
int pp = 0;
int KM1 = 0;
int KM2 = 0;
int KM3 = 0;
int c = 0;

void setup()
{
  pinMode(2, INPUT); //entrada pm
  pinMode(3, INPUT); //entrada pp
  pinMode(4, OUTPUT); // salida KM1
  pinMode(5, OUTPUT); // salida KM2
  pinMode(6, OUTPUT); // salida KM3
}

void loop()
{
  if (digitalRead(2)==HIGH)
  {
    pm=1;
  }

  //etapa 1
  if (x0 == 1 and pm == 1)
  {
    x1 = 1;
    x0 = 0;
    pm = 0;
  }

  //etapa2
  if (x1 == 1 and T1 == 1)
  {
    x2 = 1;
    x1 = 0;
    T1 = 0;
  }

  //etapa3
  if (x2 == 1 and T2 == 1)
  {
    x3 = 1;
    x2 = 0;
    T2 = 0;
  }

  //etapa4
  if (x3 == 1 and T3 == 1)
  {
    x4 = 1;
    x3 = 0;
    T3 = 0;
  }
}

```

```
//etapa0
if(digitalRead(3)==HIGH)
{
  pp=1;
}
if ((x1 == 1 or x2 == 1 or x3 == 1 or x4 == 1) and pp == 1)
{
  x0 = 1;
  x1 = 0;
  x2 = 0;
  x3 = 0;
  x4 = 0;
  T1 = 0;
  T2 = 0;
  T3 = 0;
  pp= 0;
  //KM1 = 0;
  digitalWrite(4,LOW);
  //KM2 = 0;
  digitalWrite(5,LOW);
  //KM3 = 0;
  digitalWrite(6,LOW);
}

//etapa de inicio
if (x0 == 0 and x1 == 0 and x2 == 0 and x3 == 0 and x4 == 0)
{
  x0 = 1;
}

//etapa de potencia KM1 y T1
if (x1 == 1)
{
  // KM1 = 1;
  digitalWrite(4,HIGH);

  c = c + 1;
  if(c == 300)
  {
    T1 = 1;
    c = 0;
    //digitalWrite(4,LOW);
  }
}

//etapa de potencia KM1, KM2 y T2
if (x2 == 1)
{
  //KM1 = 1;
  digitalWrite(4,HIGH);
  KM2 = 1;
  digitalWrite(5,HIGH);
  c = c + 1;
  if(c == 2700)
  {
    T2 = 1;
    c = 0;
    digitalWrite(4,LOW);
  }
}
```

```

    digitalWrite(5,LOW);
  }
}

//etapa de potencia KM2 yT3

if (x3 == 1)
{
  KM2 = 1;
  digitalWrite(5,HIGH);
  c = c + 1;
  if(c == 300)
  {
    T3 = 1;
    c = 0;
    digitalWrite(5,LOW);
  }
}

//etapa de potencia KM2 y KM3

if (x4 == 1)
{
  KM2 = 1;
  digitalWrite(5,HIGH);
  KM3 = 1;
  digitalWrite(6,HIGH);
}

fin:

//base de tiempos en milisegundos
delay(1);
}

```

Se ha programado siguiendo la secuencia del GRAFCET. Las condiciones de transición para pasar de una etapa a otra se han programado utilizando la instrucción "if". Los temporizadores se inician en la parte de potencia. También se programan en la parte de potencia y para que sea operativo en cualquier momento la actuación del pulsador de paro, los temporizadores se programan a partir de contadores con una base de tiempos prefijada en 1 ms. Finalmente en la etapa de control correspondiente se deben de poner a cero.

IMPLEMENTACIÓN EN PROTEUS

Se realiza la comprobación de las secuencias en Proteus comprobando el cronograma del arranque estrella triángulo, según se muestra en la Figura 15.

Se realiza la comprobación pulsando el pulsador de marcha y comprobando las salidas indicadas por los Leds correspondientes. Estos indican la activación de los correspondientes contactores según se muestra en la Figura 16.

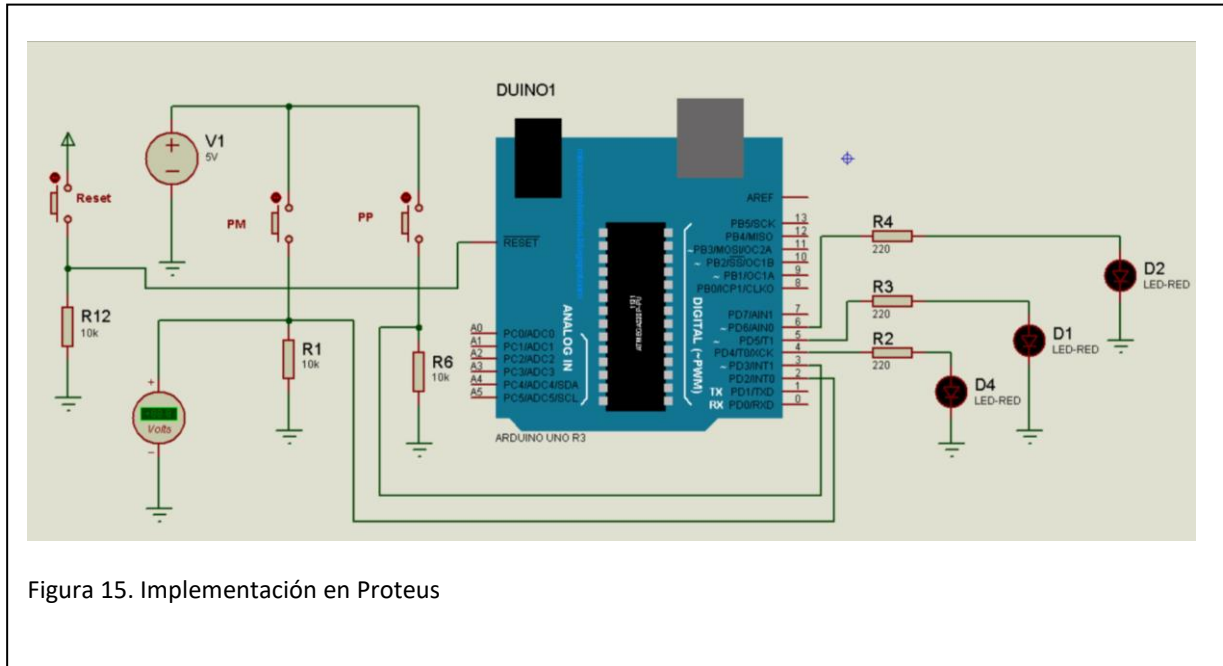


Figura 15. Implementación en Proteus

ESQUEMA DEL CIRCUITO DE CONTROL

El esquema completo del circuito de control es el mostrado en la Figura 16.

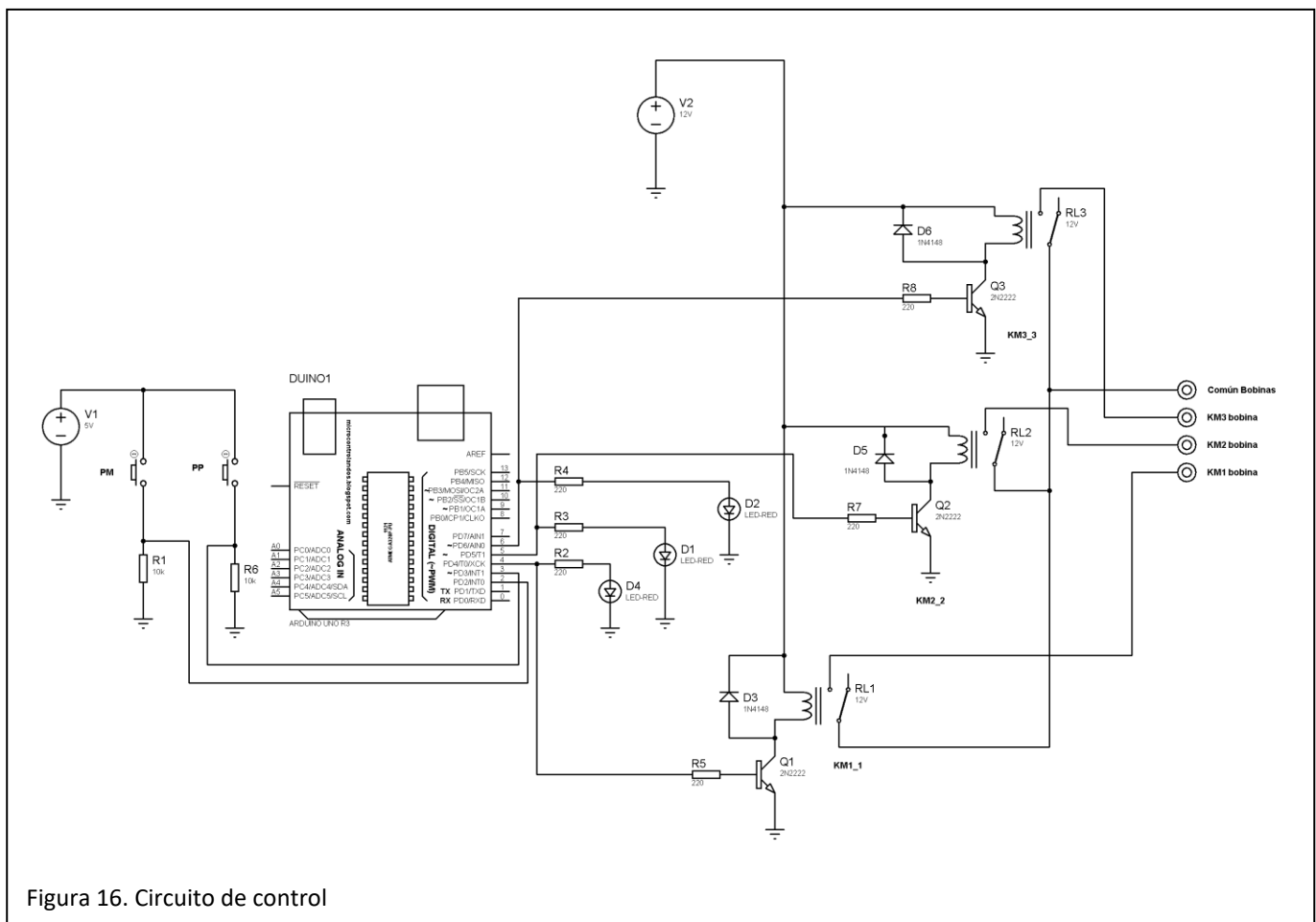
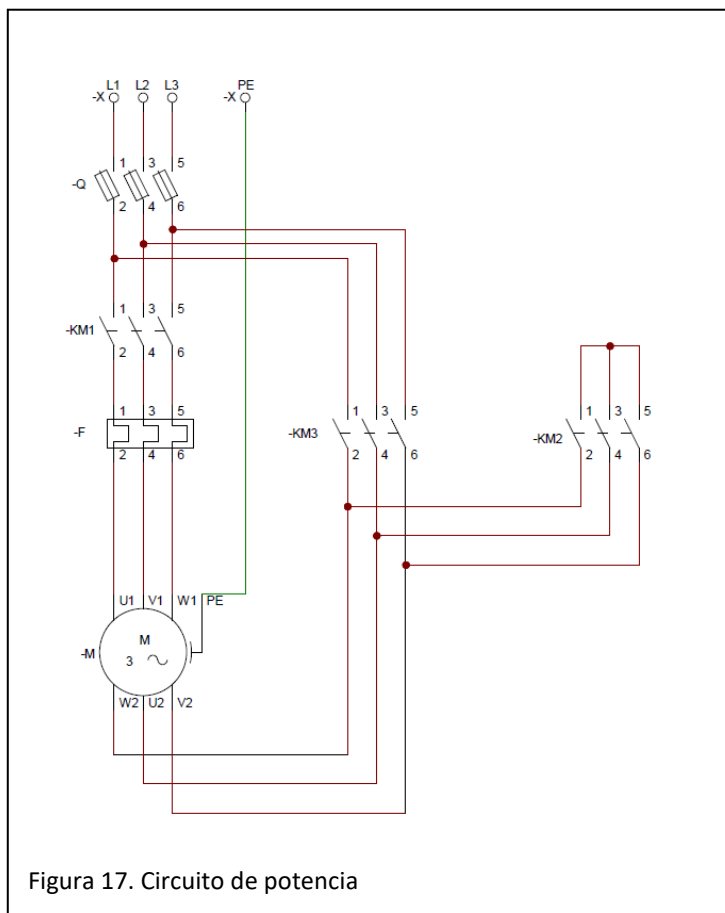


Figura 16. Circuito de control

Cuando se pulsa sobre el pulsador de marcha se inicia la secuencia de arranque. En primer lugar se activa el contactor estrella KM1. 300 milisegundos después se activa el contactor de línea KM2. Una vez transcurridos los 3 segundos programados, se abre el contactor de estrella KM1 y se cierra el contactor de triángulo KM3. En cualquier momento se puede detener la secuencia y por lo tanto el motor, actuando sobre el pulsador de paro.

CIRCUITO DE POTENCIA

El circuito de potencia está formado por las protecciones correspondientes, los contactores y el propio motor. Se muestra en la Figura 17 dicho circuito.



CIRCUITO COMPLETO

Está formado por la unión del circuito de control y del circuito de potencia con sus correspondientes protecciones.

CONCLUSIONES

Se ha implementado un arranque estrella triángulo a partir del GRAFCET obtenido del cronograma de funcionamiento. Esta automatización mostrada en el GRAFCET se ha implementado en "C" programando la secuencia en Arduino. Se ha mostrado cómo realizar la transferencia de la secuencia de control al lenguaje de programación en "C" y se ha mostrado su funcionamiento realizando su implementación y comprobándose la similitud entre la programación por contactos y la programación en un lenguaje de alto nivel. Se ha comprobado la similitud por la analogía entre los contactos serie y paralelo con las instrucciones "if" y "or" en programación.