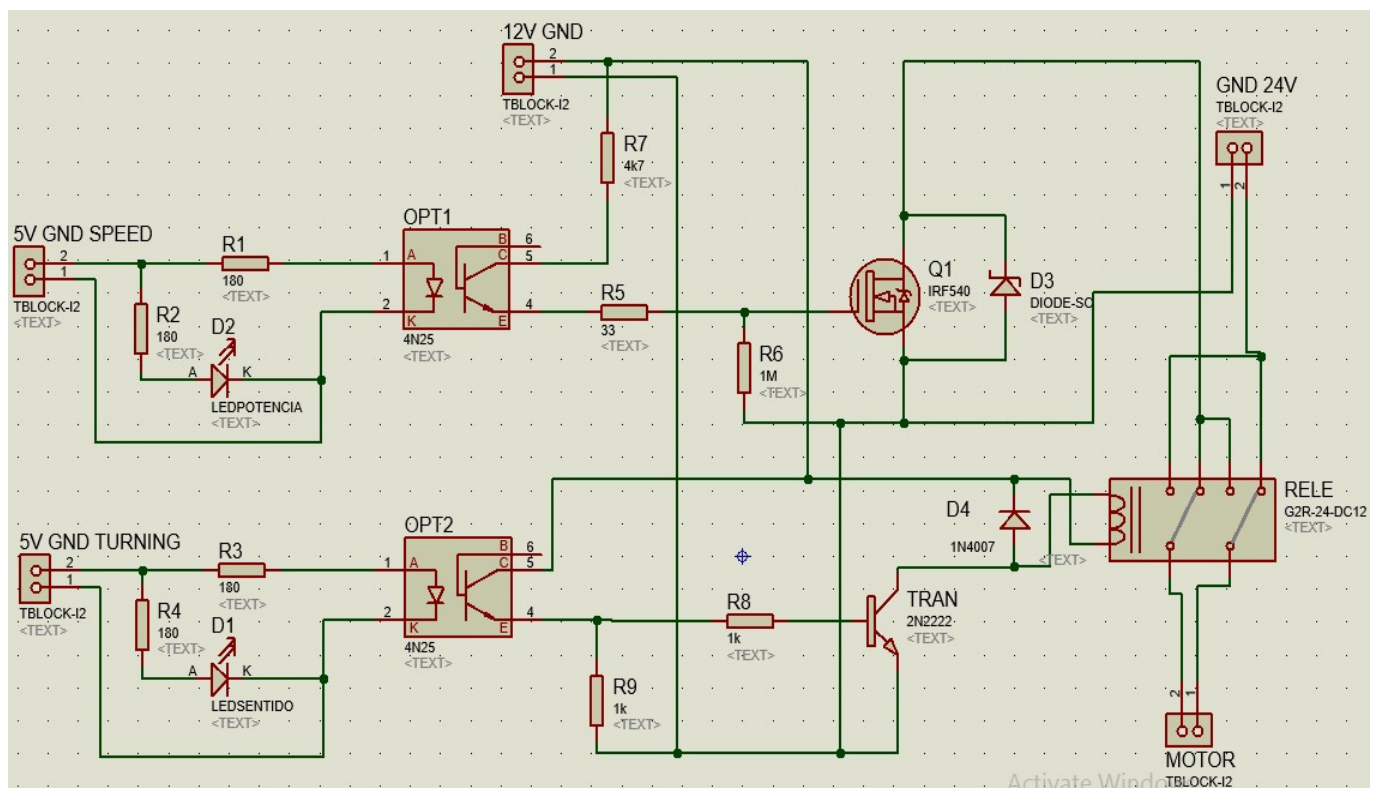


PRÁCTICA 11		PLACA DE CIRCUITO IMPRESO	
		CONTROL DE MOTOR DC CON MOSFET IRF540	
Apellidos	PEÑALVER FERNÁNDEZ		
Nombre	ADRIÁN		
Grupo	1º MET	Curso	2018/2019
Fecha de entrega			

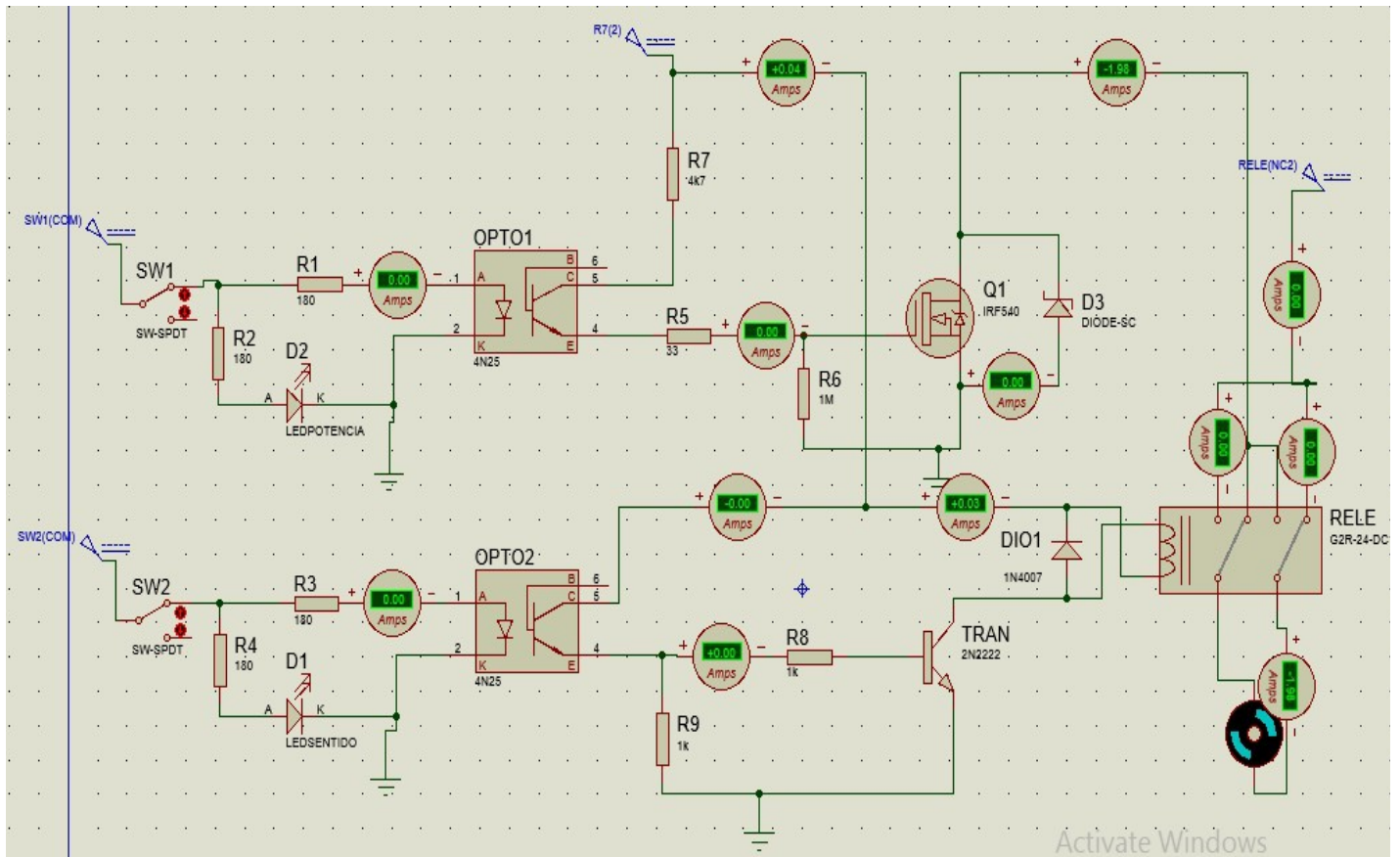
1. Dibuja el esquema con Proteus. Documentalo.



Aquí podemos ver la diferencia entre la etapa de potencia y la de control, en la etapa de control, las tierras pueden ir juntas o separadas, mientras que en la de potencia deben ir unidas a la fuerza.

Es muy importante a la hora de la realización del esquema, separar la parte del circuito que pertenece a la etapa de control de la etapa de potencia.

2. Realiza la simulación del circuito. Documentála.



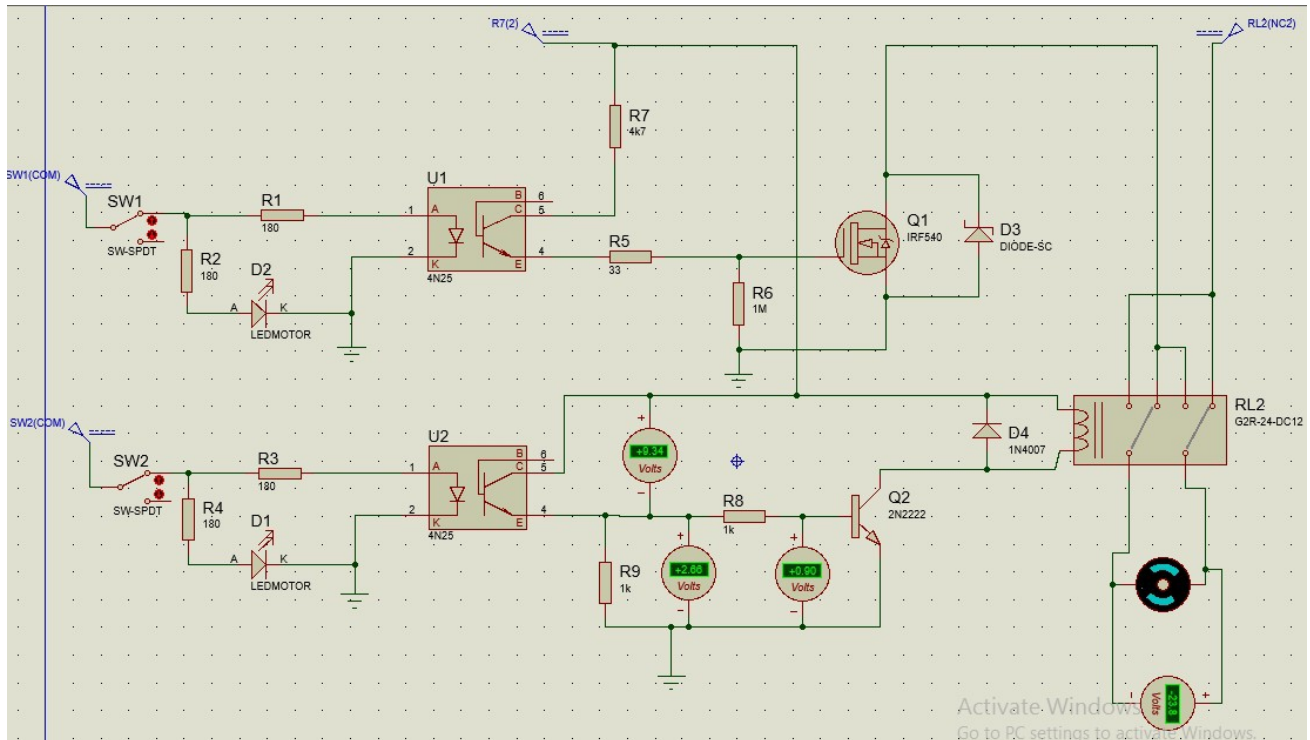
En esta captura se ve el circuito con diferentes amperímetros a lo largo de todo el circuito, destacan en este circuito, el amperímetro en el motor que marca 1,98 A, en el resto de amperímetros el valor es mucho mas bajo.

Esta corriente mucho mas elevada es gracias al MOSFET que nos permite elevarla para controlar mucha mas potencia, como en este caso un motor.

Cuando se activa el SW1 el motor comienza a girar mientras que si se abre el motor se para.

Con el SW2 si se cierra se excita el relé y si se queda abierto no se excita, de esta forma se consigue cambiar el sentido de rotación del motor.

Aunque el control de la velocidad del motor debe permitir hacer rampas ascendentes y descendentes de velocidad, esto no será posible simularlo en Proteus, puesto que la simulación en este caso es todo o nada.



En esta imagen vemos los voltímetros, que en el caso del motor también vemos como cambia el signo y eso significa que se cambia el sentido de rotación. Si se abre el interruptor SW2 entonces la diferencia de potencial pasaría de ser de 9,34 a 12 (V), puesto que el optocoplador está abierto y el transistor también.

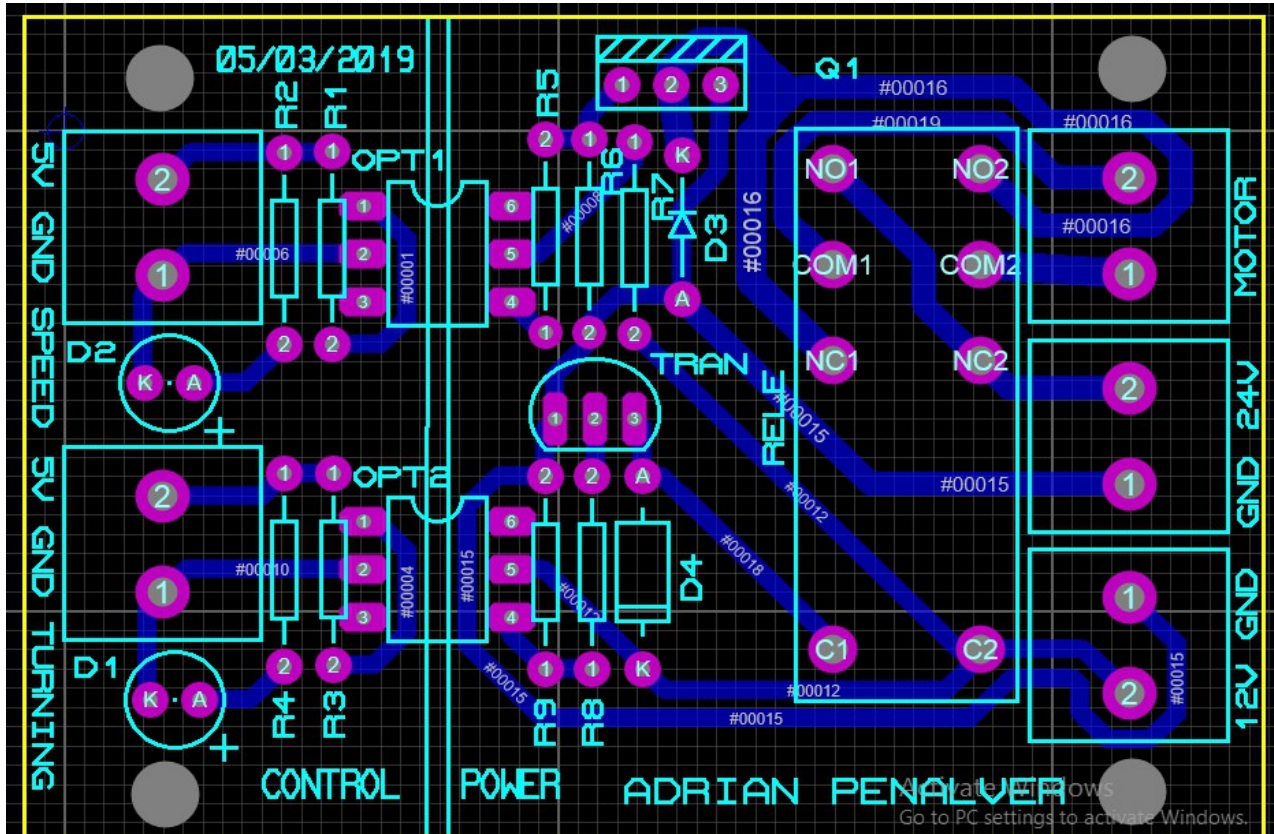
3. Explica el funcionamiento del circuito.

El circuito consiste en una etapa de control compuesta por dos circuitos con una entrada digital a 5 (v) de Arduino, una vez que se pone a nivel alto la entrada, entonces el optocoplador cierra el circuito en la etapa de potencia, además se enciende un LED para indicarnos si tenemos un nivel alto o bajo.

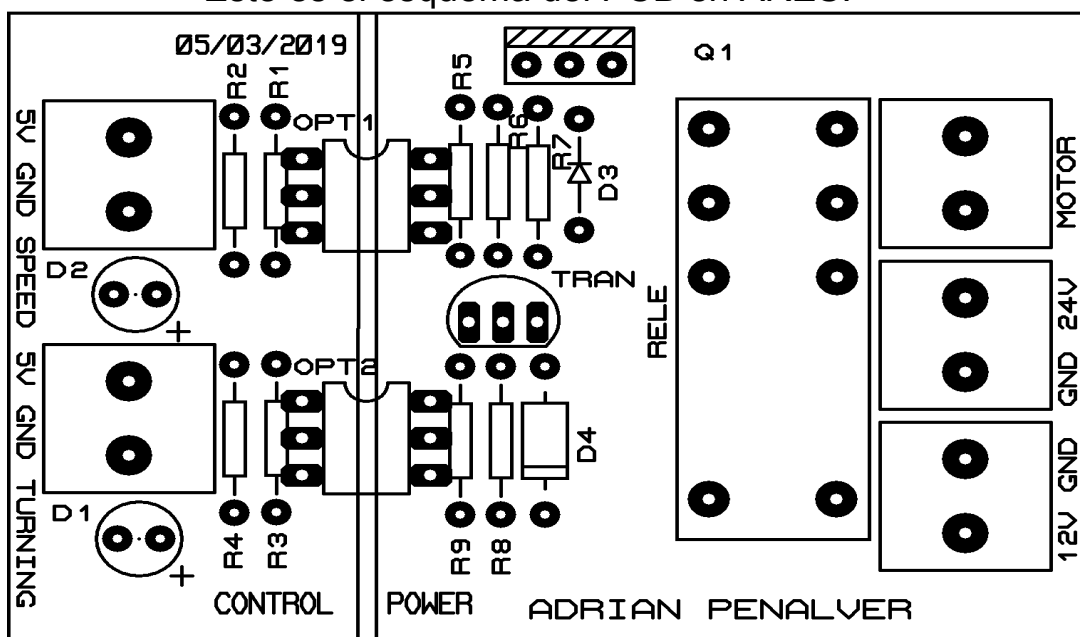
Por otra parte tenemos la etapa de potencia, esta etapa está alimentada a 12(v), si la etapa de control que se encarga del control de giro se pone a nivel alto, entonces el optocoplador cierra el circuito, se activa el transistor y una corriente recorre la bobina excitándola y haciendo cambiar los contactos en el relé y cambiando así el sentido de giro.

Por otro lado si se pone a nivel alto la etapa de control de velocidad, entonces el optocoplador cierra el circuito que controla la velocidad, se crea un potencial en la puerta del MOSFET, que crea una corriente que permite alimentar el motor a 24 (V) máximo y gracias a la señal PWM que enviamos con el Arduino, se puede controlar la velocidad jugando con la tensión en la puerta del MOSFET, ya que con pequeñas variaciones de voltaje en la puerta de este, se puede controlar la corriente que entrega al motor y de esta forma la velocidad.

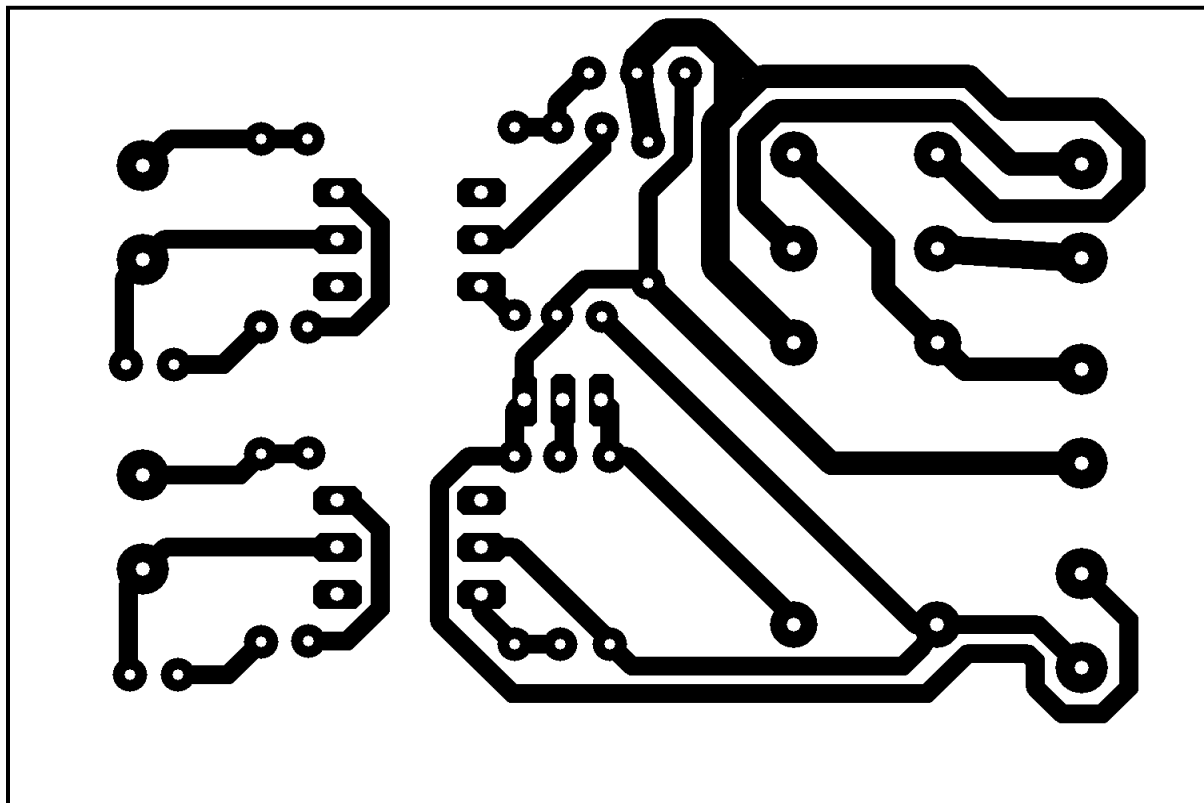
4. Obtén el PCB. Adjunta el archivo de Proteus, la impresión de las pistas y la de los componentes.



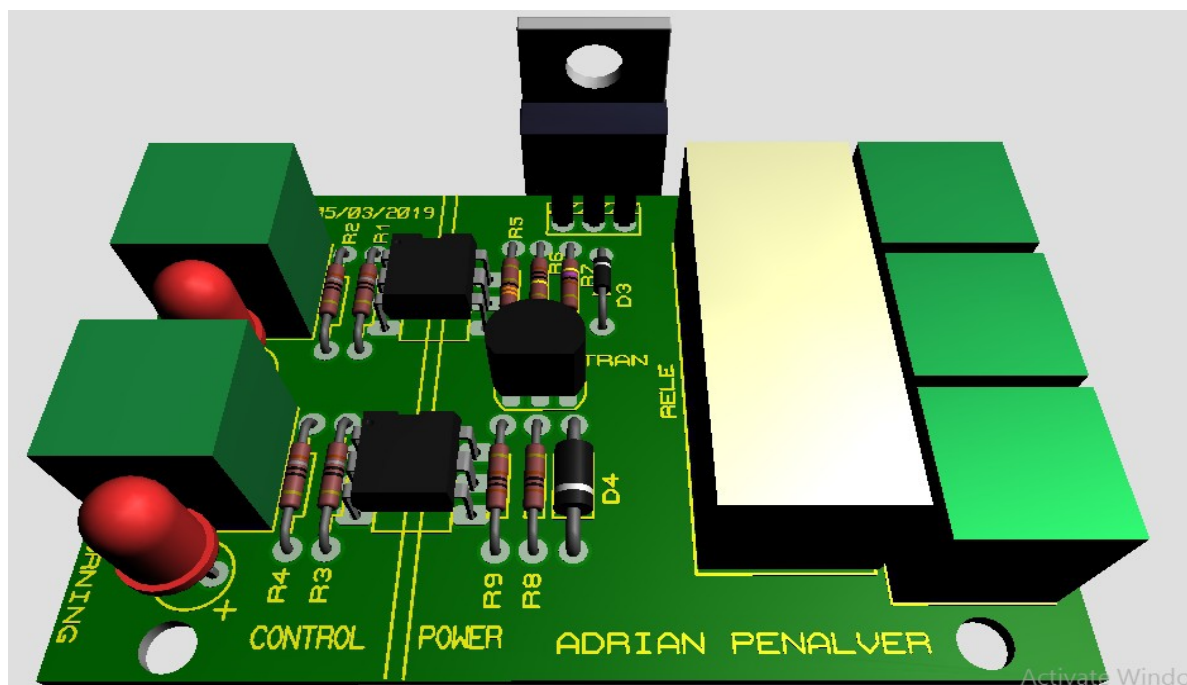
Este es el esquema del PCB en ARES.



Esta cara es la de los componentes.

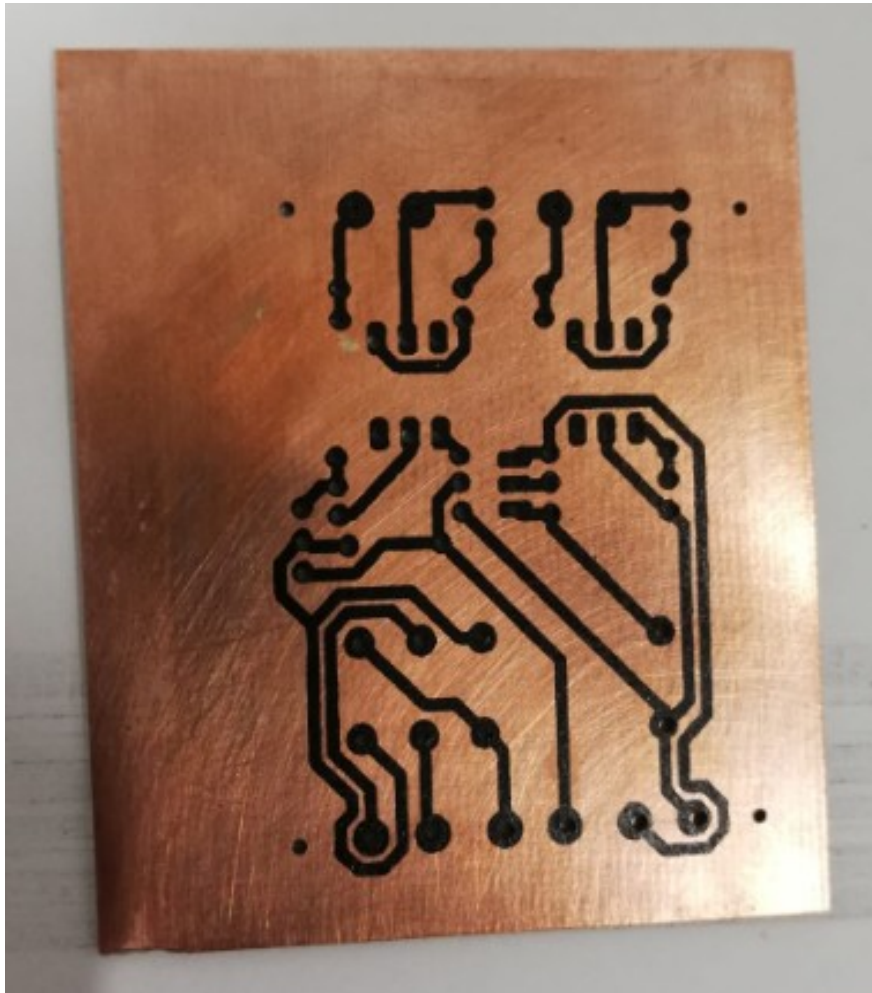


En esta imagen se ve la parte del cobre.

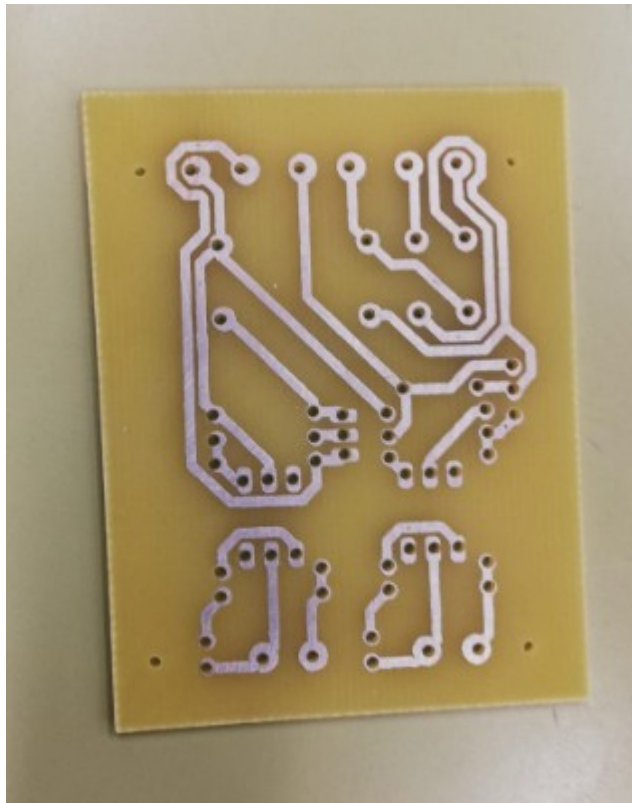


Una imagen 3d de la placa con los componentes.

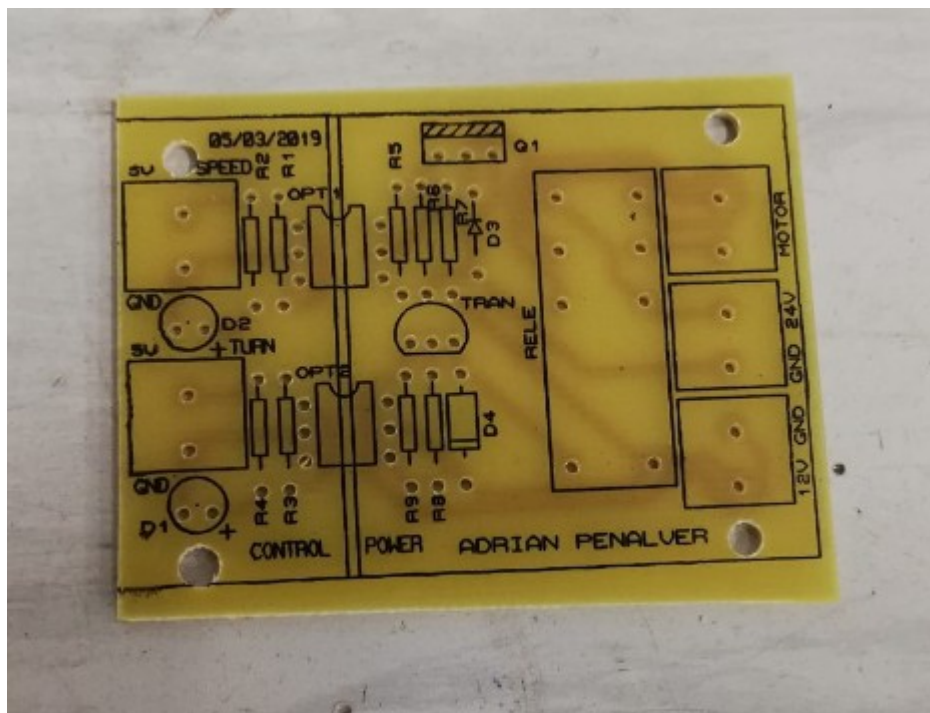
- 5. Realiza la placa de circuito impreso por el método del planchado o papel amarillo (pistas), utilizando la fresadora para los taladros. Muestra físicamente el resultado final.**



PISTAS EN LA CARA DEL COBRE.

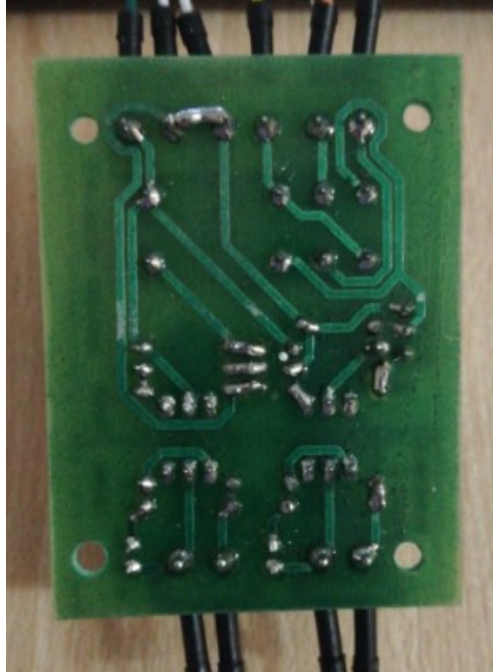


PISTAS UNA VEZ PASADAS POR EL ACIDO Y ESTAÑADAS.



PARTE SUPERIOR CON SERIGRAFÍA.

6. Añade los componentes electrónicos a la placa. Copia imágenes de la placa final. Muestra físicamente el resultado final.

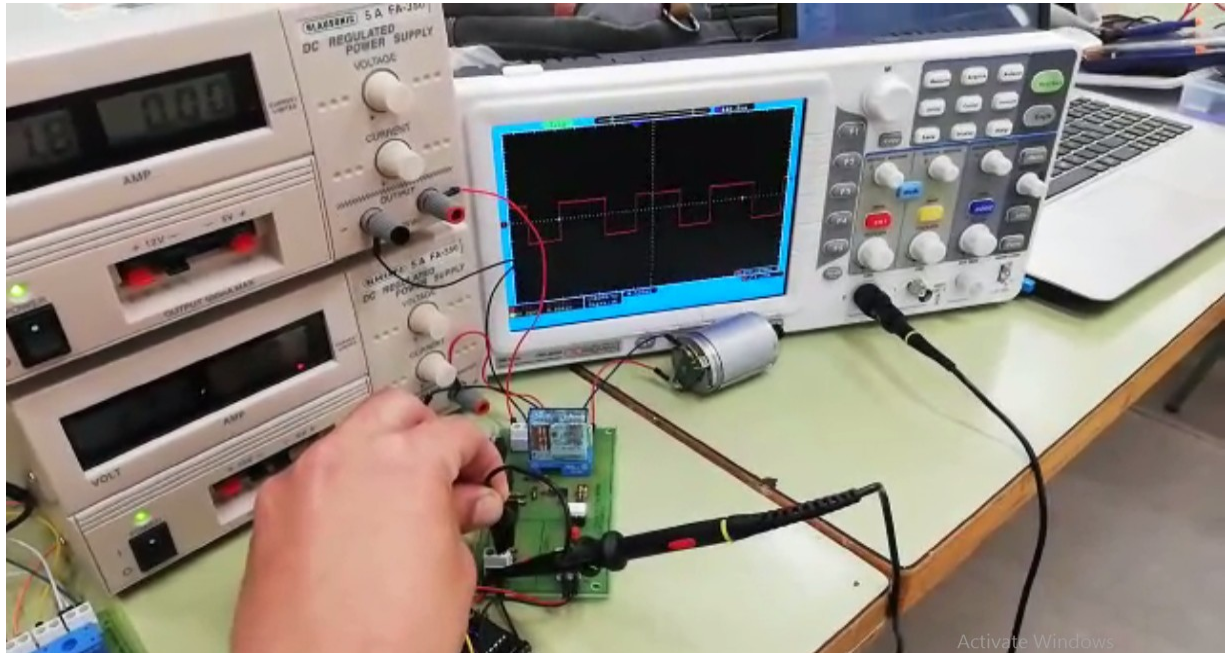


PARTE POSTERIOR CON COMPONENTES SOLDADOS Y CON TINTE VERDE.

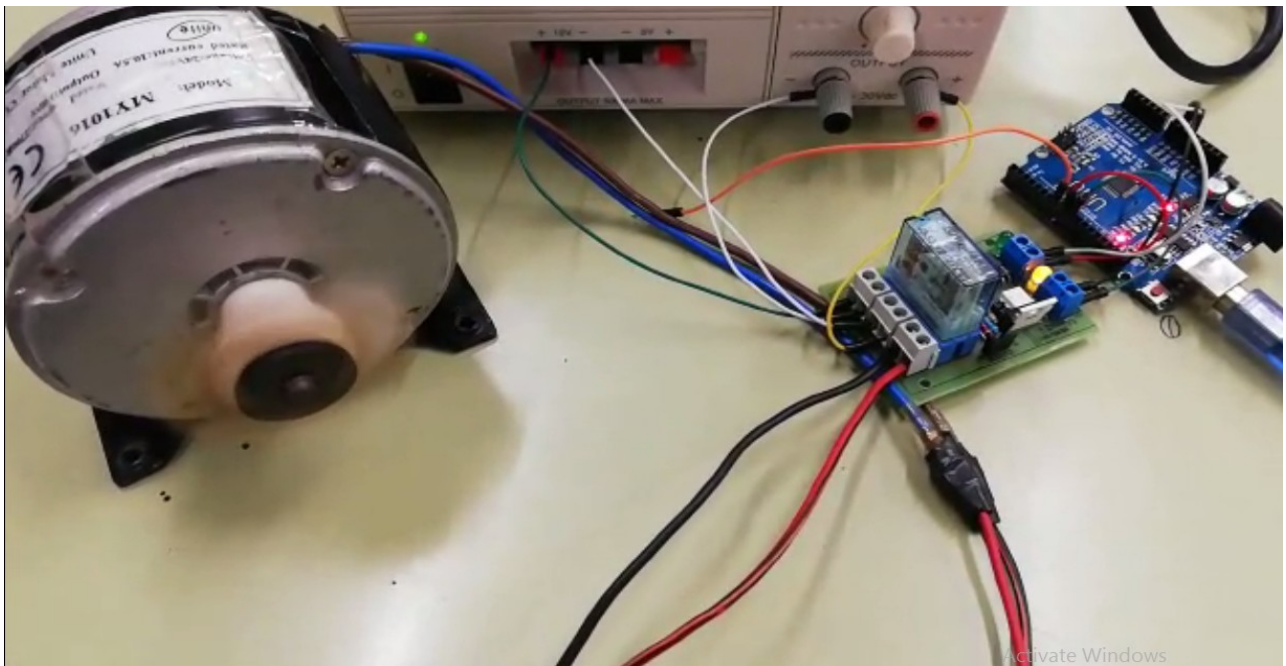


PARTE SUPERIOR DEL CIRCUITO CON COMPONENTES SOLDADOS.

7. Utiliza la placa obtenida para modificar el sentido de giro de un pequeño motor. Muestra "in situ" el correcto funcionamiento e inserta alguna imagen.



En esta imagen se ve una prueba de la placa sin carga, donde se puede ver perfectamente la señal PWM en el osciloscopio, gracias a esta prueba, nos dimos cuenta que la resistencia de 1M era demasiado grande y se tenía que cambiar por otra mas pequeña, en este caso de 100k



En esta imagen, la placa funciona con un motor y el programa en el Arduino realiza una rampa ascendente, luego otra descendente y finalmente un cambio de sentido cuando el motor esta parado para otra vez volver a repetir el ciclo.

El programa utilizado para que funcione correctamente es el siguiente:

```
#define PIN_MOTOR 9

#define PIN_SENTIDO 8
#define PIN_ONOFF 7

// Declaracion de variables enteras.

int sentidoGiro;
int i = 0;
int valor;

void setup()
{
    pinMode(PIN_MOTOR,OUTPUT); // Declara el pin 9
    de Arduino como SALIDA
    pinMode(PIN_SENTIDO,OUTPUT);

    Serial.begin (9600); // Establece la velocidad de
    datos en bit por segundo (baudios) para la transmisión de datos en serie.
}

void loop()
{
    while(digitalRead(PIN_ONOFF) == HIGH) // Mientras el pin
    7 esté en estado ALTO, ejecuta las sentencias que hay dentro de las llaves.
    (ON/OFF)
    {
        for (i=0; i < 160; i+=1) // RAMPA DE ACELERACIÓN 160
        antes 254
        {
            analogWrite(PIN_MOTOR, i); // Escribe el valor de la
            variable i en el pin analógico 9 (motor).
            Serial.println(i); // Imprime los datos en el
            puerto serie, seguido de un retorno de carro y de línea nueva.
            delay(70); // Hace una pausa en el
            programa durante el tiempo (50 milisegundos), bloqueando todo el programa durante
            dicho tiempo.
        }
        delay (2000);

        for (i=159; i > 1; i-=1) // RAMPA DE DECELERACIÓN
        antes 255 ahora 159
        {
            analogWrite(PIN_MOTOR,i); // Escribe el valor de la
            variable i en el pin analógico 9 (motor).
            Serial.println(i); // Imprime los datos en el
            puerto serie, seguido de un retorno de carro y de línea nueva.
            delay(70); // Hace una pausa en el
            programa durante el tiempo (50 milisegundos), bloqueando todo el programa durante
            dicho tiempo.
        }
        analogWrite(PIN_MOTOR,0);
        delay(4000);

        if(sentidoGiro == 0){
```

TPMMEE

Calle Grecia ,56 | 30203 Cartagena
Tlf. 968120909 | Fax 968500077
30001801@murciaeduca.es
politecnicocartagena.com

Código Pendiente | Revisión: 00 | Página 11 de 11

```
        sentidoGiro = 1;
        digitalWrite(PIN_SENTIDO, HIGH);
    }else{
        sentidoGiro = 0;
        digitalWrite(PIN_SENTIDO, LOW);
    }
}
}
```