

# PROYECTO CDI y ESTROBOSCOPICO con ARDUINO

El proyecto trata de regular los grados en los que debe saltar el encendido (la chispa de la bujía) de una motocicleta, en función de las revoluciones a las que vaya el motor.

Para controlar el encendido, utilizaremos **Arduino**, al que enviaremos la señal del **PickUp** por un Pin Analógico y monitoreará la salida de este con la cual **determinaremos las RPM** a las que gira el motor.

**Crearemos unas condiciones** en función de las RPM's y dentro de estas colocaremos unos **DELAYS que incrementando o decrementando los Milisegundos causarán la regulación** de la chispa del encendido en función de las RPM's a las que vaya el motor, y con ayuda de **“una lampara estroboscópica”** y pegando un disco con una marca justo donde está la “leva”, podremos determinar si la chispa, está saltando en el momento que nosotros queremos.

Dado que Arduino no puede suministrarnos potencia suficiente para activar la bobina debido su poca potencia de salida, nos veremos obligados a utilizar un MOSFET o SCR, para poder manejar la tensión que necesita la bobina con la velocidad de corte y saturación necesarias para no causar retardos que puedan hacer inviable este proyecto.

## **Para realizar este proyecto necesitaremos:**

- Sensor Pick Up.
- Motor DC. (al que pegaremos una “leva”).
- Mosfet N, o SCR X0405MF.
- LED's (con sus resistencias) para crear efecto estroboscópico.
- Arduino o ATMEGA328P.

## **\*No Necesarios, pero recomendables:**

- Bobina de descarga de 12v DC.
- Bujía.

# Montaje y Conexionado

## Sensor PickUp (2 “pines”):

- Data => a Pin Analógico 0 (A0). ARDUINO
- MASA (Muy importante). => Masa.

## -SCR X0405MF ( 3 “pines”): (mirándolo de frente y comenzando por la izq).

- Source => Hacia la bobina o led y de esta a NEGATIVO(de su fuente).
- Drain => +12V.
- Gate => a Pin Digital 9 (9). ARDUINO

## - Bobina de Descarga: (3 “pines” + MASA(en el cuerpo de esta)) o LED\* (2 “pines”)

- Positivo\* => Patilla SOURCE del SCR.
- Negativo\* => NEGATIVO de la fuente de 12V.
- SALIDA ALTA => POSITIVO de la BUJÍA.
- MASA(Cuerpo) => a MASA de la BUJÍA, NO EN LA FUENTE DE 12V.

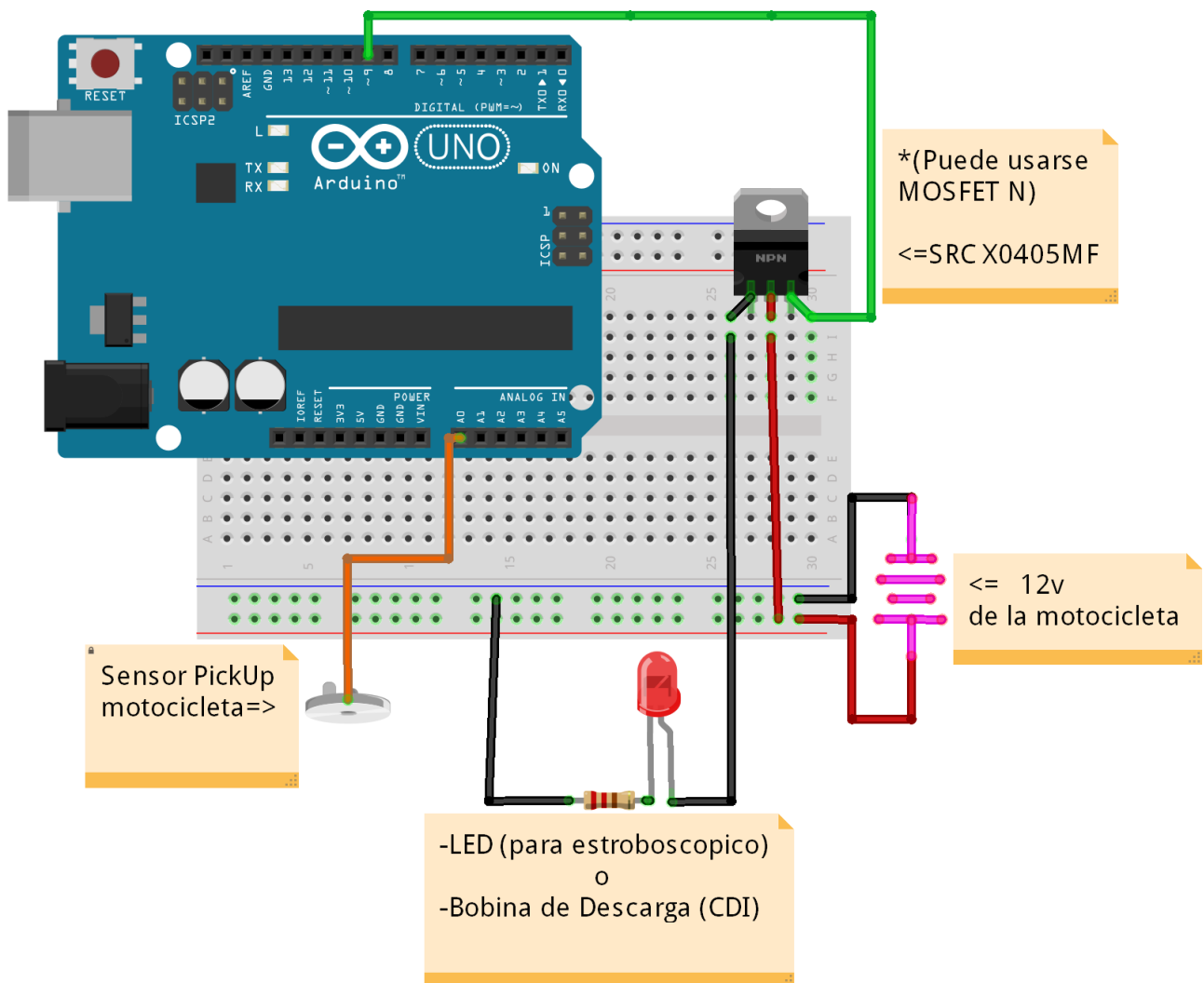
## - Bujía (2 “pines”):

- Positivo => SALIDA ALTA de la BOBINA de Descarga.
- Masa => a MASA de la BOBINA de Descarga.

El plano de montaje:

**Está montado con el LED**, para producir el efecto estroboscópico.

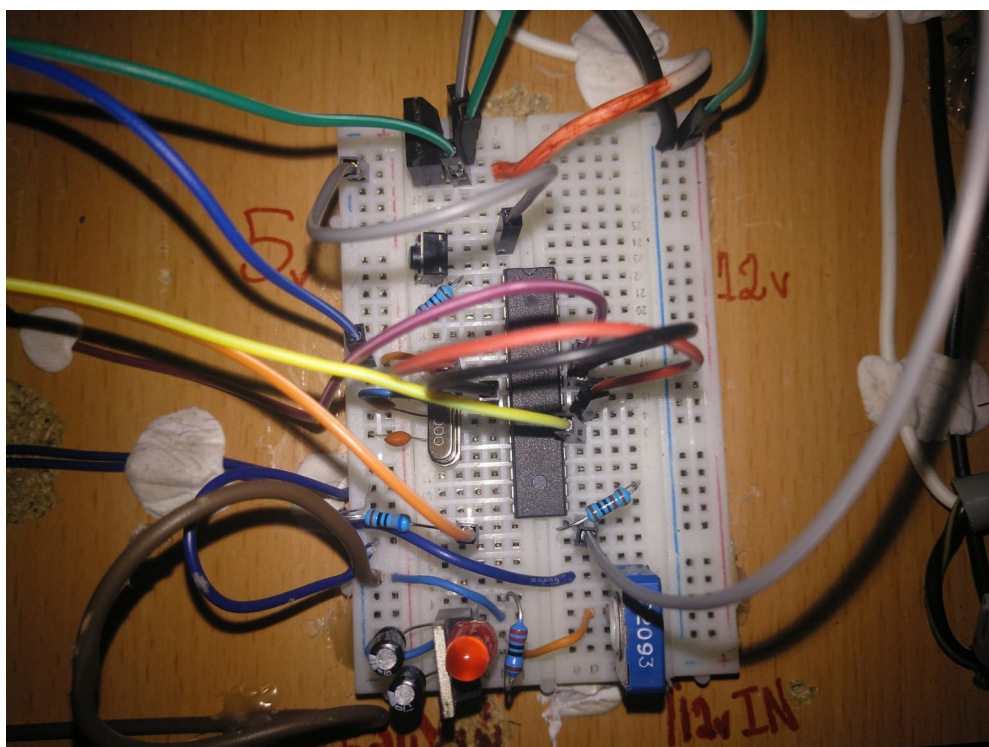
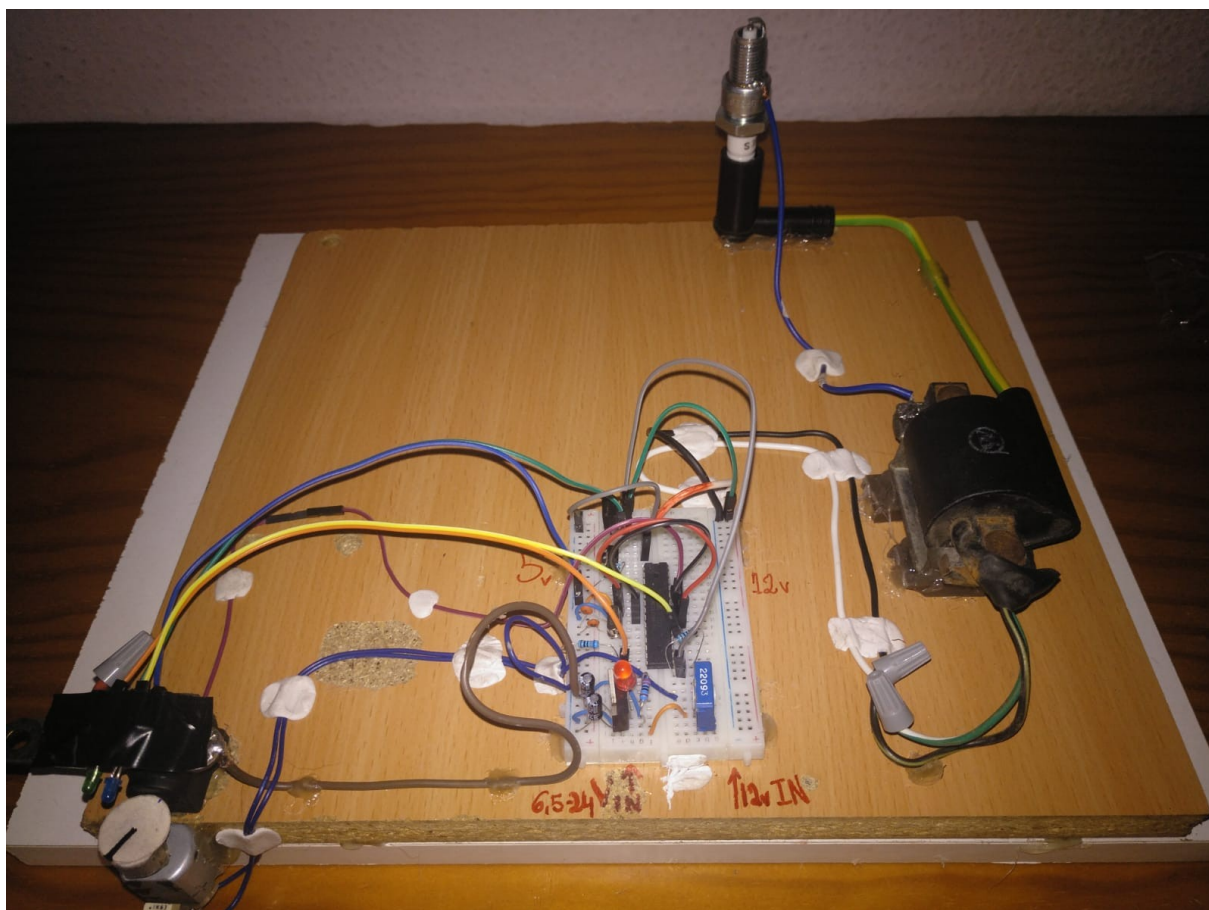
**\*En caso de querer conectar la BOBINA**, hacerlo cambiando el LED por la BOBINA y conectando la BUJÍA a esta como está anteriormente explicado.



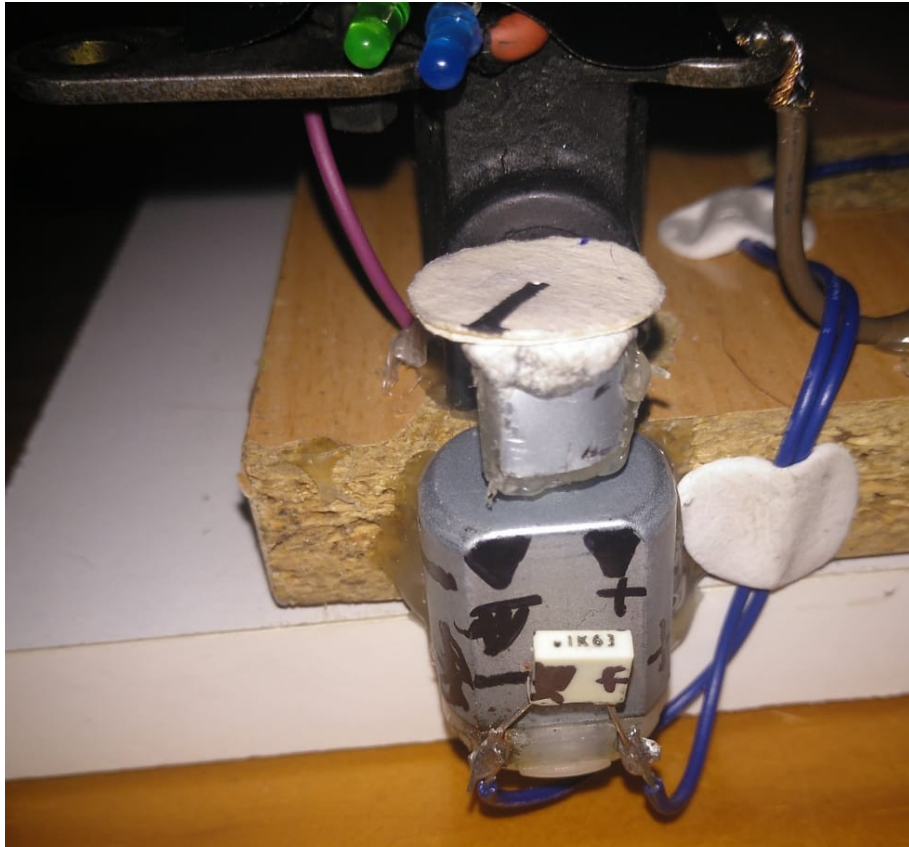
fritzing

Todo es muy bonito en el papel... En mi caso, lo he montado utilizando un ATMEGA328P sobre una BreadBoard y con la bobina, bujía, sensor Pickup, motor DC de 3-6v (al que le he pegado la leva, cumpliendo con las distancias entre esta y el sensor, según el fabricante) y LED para obtener el efecto estroboscópico.

El resultado ha sido el siguiente:



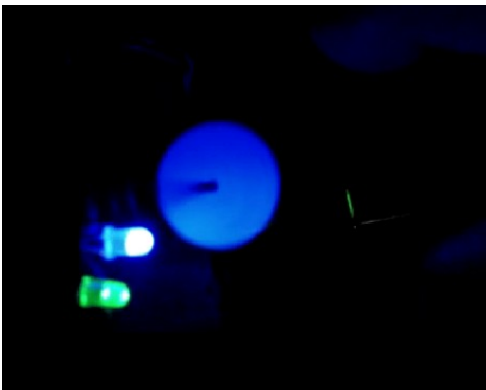
El motor, leva y sensor PickUp visto de más cerca:



Aquí podemos observar la distancia entre la leva y el sensor (0.8mm) :



Por último, mostrar un par de imágenes de nuestro efecto estroboscópico en funcionamiento:



## CODIGO ARDUINO

```
const int sensorPin = A0; // select the input pin for the potentiometer
const int ledPin = 9;    // select the pin for the LED
float outputValue = 0; // variable to store the value coming from the sensor
int sensorValue = 0;
float freq = 0;
const int Pulsador = 7; //Agregando Pulsador para control de DELAYS

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
  pinMode(Pulsador, INPUT);
} // setup()

void loop()
{
  // Lectura del potenciómetro y mapping
  sensorValue = analogRead(sensorPin);
  outputValue = map(sensorValue, 0, 1023, 0, 50);

  //creando condiciones para realizar los retardos
  int PulsadorCount = 0;

  if (digitalRead(Pulsador) != 0){

    if (Pulsador >= 8){
      PulsadorCount = 0;
    }

    PulsadorCount ++;

  }

  freq = (1000)/(outputValue); //ESTA LINEA ESTA MAL
```



```

if (outputValue > 0)
{
    //Encendemos el led

// estos DELAYS juntos causan un retardo de una vuelta completa (16mS)
/*
    delay(2);
    delay(2);
    delay(2);
    delay(2);
    delay(2);
    delay(2);
    delay(2);
    delay(2);
    delay(2);*/
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);

```

//PRUEBA RETARDOS CON CONDICIONES

```

if (Pulsador == 1){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(1);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}

else if (Pulsador == 2){

    //retardo de 2ms para atrasar/adelantar la chispa)
    delay(2);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}

else if (Pulsador == 3){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(3);

```

```

//Prende la chispa
digitalWrite(ledPin, HIGH);
// Esperamos 1 ms para que el encendido sea visible
delay(1);
// Apagamos el led
digitalWrite(ledPin, LOW);
Serial.println(outputValue);
Serial.println(60*freq);
delay(outputValue);
}

else if (Pulsador == 4){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(4);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}

else if (Pulsador == 5){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(5);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}

else if (Pulsador == 6){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(6);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}

else if (Pulsador == 7){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(7);

```



```

//Prende la chispa
digitalWrite(ledPin, HIGH);
// Esperamos 1 ms para que el encendido sea visible
delay(1);
// Apagamos el led
digitalWrite(ledPin, LOW);
Serial.println(outputValue);
Serial.println(60*freq);
delay(outputValue);
}

else if (Pulsador == 8){

    //retardo de 1ms para atrasar/adelantar la chispa)
    delay(8);

    //Prende la chispa
    digitalWrite(ledPin, HIGH);
    // Esperamos 1 ms para que el encendido sea visible
    delay(1);
    // Apagamos el led
    digitalWrite(ledPin, LOW);
    Serial.println(outputValue);
    Serial.println(60*freq);
    delay(outputValue);
}
}

} // loop()

```

ESTO ES TODO. SOLO FALTA IR “AFINANDO EL CÓDIGO” Y ELIMINAR TODOS LOS DELAYS SUSTITUYENDOLOS POR MILLIS, SUERTEE!!!