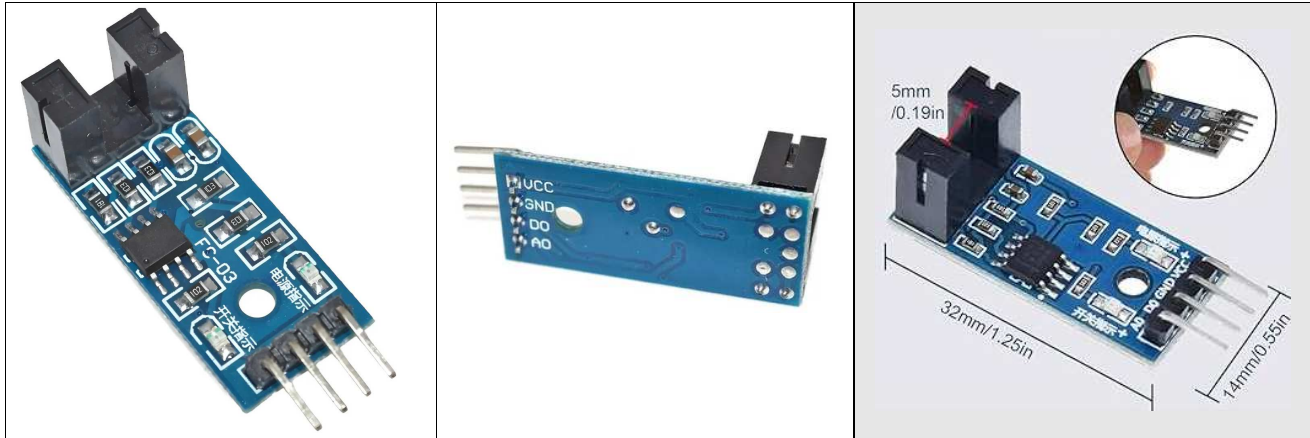


Tacómetro Lm393 Sensor Optico Horquilla



Módulo/Tacómetro: Este sensor de velocidad, chip LM393, tiene un voltaje de trabajo CC que oscila entre 3.3 - 5 V, y el comparador nos brinda una señal de salida del limpia, con buena forma de onda, y tiene capacidad de conducción, y un muy bajo consumo.

Aplicaciones:

Se utiliza comúnmente en sistemas de medición de velocidad, como tacómetros de motor, encoders ópticos, o para digitalizar la salida de sensores resistivos en diversas aplicaciones.

Con este sensor, podrás contar las veces que un objeto interrumpe el as de luz que pasa por la "U" (el optoacoplador). Entonces si quieres medir las revoluciones por minuto de un eje que gira, deberás adaptar tu dispositivo para que en cada vuelta del eje corte por un instante el as de luz. Verifica que la velocidad de rotación (una vuelta) le de el suficiente tiempo al sensor para realizar la lectura (La cantidad de rayos de la rueda puede ser desde uno en adelante). Pero si fuera muy rápida la rotación, podrías mediante una relación de engranajes, hacer que, por ejemplo, cada 4 vueltas del eje, la rueda corte o genere una sola interrupción del sensor. Luego la velocidad sería 4 veces mayor; para este ultimo caso podrías tomar los engranajes de una impresora inservible o comprarlos en casas especializadas en autos RC (autos a escala manejados por control remoto) etc.



Podemos encontrar placas con optoacoplador por unos por 3 a 4 Dólares en vendedores Argentinos.

No son componentes fáciles de encontrar porque **cada vendedor le pone un nombre diferente** y las traducciones de estas páginas son bastante malas. Probar buscando "Tacometro Lm393" o "photo interrupter" o "sensor velocidad Arduino" o "sensor ranurado Arduino"

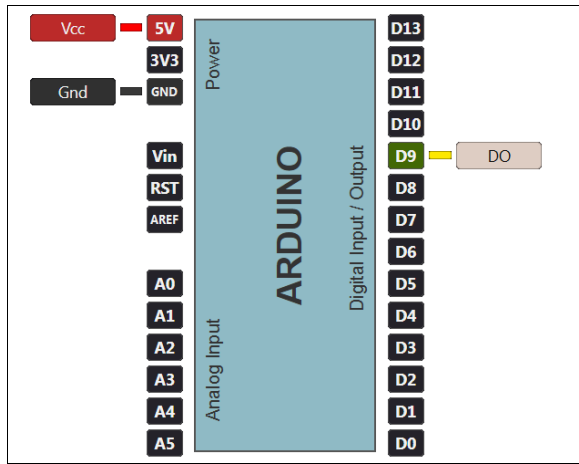
Esquema eléctrico

Si usás esta placa comercial, que como hemos dicho en general es recomendable, el montaje de un optointerruptor a Arduino es realmente sencillo. Alimentamos el módulo a través de Vcc y GND conectándolos, respectivamente, a la salida de 5V y GND en Arduino.



Por otro lado conectamos la salida digital del sensor a una entrada digital para leer el estado del sensor.

Tacómetro Lm393 Sensor Optico Horquilla



No he realizado pruebas conectando el pin **analógico**, pero es cuestión de investigar un poco y hacer pruebas.

Para conectarlo con Arduino, el esquema eléctrico es muy simple, ya que necesitamos alimentar el módulo correctamente, respetando el esquema del componente.

Como ves, el sentido de alimentación de ambas ramas suele estar invertido. Verificar el patinaje de tu optoacoplador, aunque generalmente es el mostrado para este modelo.

Ejemplos de código

Tenemos varias opciones para leer un optointerruptor con Arduino. Si estamos detectando la presencia de un objeto, simplemente leemos el estado de la entrada digital, la cual alerta cuando se interpone el objeto, pero puede realizar múltiples lecturas, las que deberás saber interpretar.

Cuando el sensor se dispara, ejecutamos las acciones necesarias, cómo incrementar un contador, o medir el tiempo entre disparos.

Código, muy simple.

```
const int PinSensor = 10;
void setup() {
  Serial.begin(9600); // Iniciar Puerto Serie
  pinMode(PinSensor , INPUT); // Defino Pin Como Entrada
}

void loop(){
  int value = 0;
  value = digitalRead(PinSensor ); // Lectura Digital del Pin que Usamos

  if (value == LOW) { // Envío Mensajes
    Serial.println("Optointerruptor activado");
  }else{
    Serial.println("Optointerruptor Desactivado");
  }
  delay(1000); // Tiempo de espera para Próxima Medición
}
```

RECOMENDACIONES PARA LA CONEXIÓN

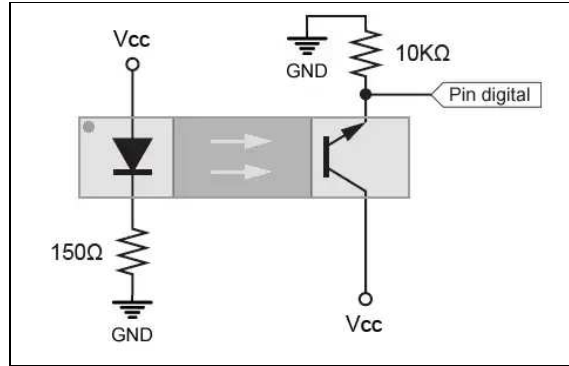
Si el elemento que activa el sensor lo hace regularmente y por un periodo de tiempo, y lo que queremos es detectar cuantas veces lo hace, entonces, la conexión que deberemos realizar deberá ser tal y como vimos cuando detectábamos el botón presionado.

Tacómetro Lm393 Sensor Optico Horquilla

Alimentamos al fotodiodo a través de una resistencia para limitar su corriente.

En la rama del fototransistor empleamos una resistencia de **pull-down** para leer el estado del sensor.

Si esta terminología te resulta incomprensible, repasa la guía del "Botón/Pulsador" en la Página de Arduino que usamos en clase (www.programacion.ar/arduino)



ACÁ Poner Protoboard Mostrando Conexiones

```
// Pull-Down Detecto Cuando el haz de luz ES Cortado
//
int PinDigital = 10; // Pin Digital donde conectare el Pin Digital
int EstadoLeido = 0;
int Activado = 1; // Inicializo como si estuviera detectando un Corte del haz de Luz

void setup( ) {
  Serial.begin(9600);
  pinMode(PinDigital, INPUT);
}

void loop( ){
  /***** Verifico si la Luz Pasa (Nada Detectado) *****/
  EstadoLeido = digitalRead(PinDigital); // Leemos el PinDigital.
  if (EstadoLeido == HIGH) { //Pregunta si la Luz Esta Pasando
    Activado = 0; // NADA ha sido detectado (Haz de luz OK)
  }
  /***** Detecto Haz de Luz Cortado *****/
  EstadoLeido = digitalRead(PinDigital); // Leemos el PinDigital.
  if (EstadoLeido == LOW && Activado == 0) { // LOW es que se corto el haz de luz
    Activado = 1; // Activo detección Haz Cortado
    Serial.println("Haz de Luz Cortado -----> OFF "); // Acá podemos contar o tomar Acción
  } else {
    Serial.println("Luz OK -----> ON");
  }

  delay(200); // Detenemos el Programa (Solo para ver bien que pasa).
} // Finaliza función loop( )
```

ACÁ Poner Protoboard Mostrando Conexiones (Pull-Up).

```
// Pull UP Detecto Cuando el haz de luz deja de ser cortado
//
int PinDigital = 10; // Pin Digital donde conectare el Pin Digital
int EstadoLeido = 0;
int Activado = 0;

void setup( ) {
  Serial.begin(9600);
  pinMode(PinDigital, INPUT);
}

void loop( ){
  /***** Detecto cuando se Sensor se Activa *****/
  EstadoLeido = digitalRead(PinDigital); // Leemos el PinDigital.
  if (EstadoLeido == LOW) { //Pregunta si el pulsador está Activado
```

Tacómetro Lm393 Sensor Optico Horquilla

```
    Activado = 1; //La variable cambia de valor
  }
  /***** Detecto cuando Sensor no esta Activo *****/
  EstadoLeido = digitalRead(PinDigital); // Leemos el PinDigital.
  if (EstadoLeido == HIGH && Activado == 1) {
    Activado = 0; //La variable vuelve a su valor original
    Serial.println("Sensor -----> ON");
  } else { // Si el valor es bajo:
    Serial.println("Sensor OFF");
  }

  delay(200); // Detenemos el Programa (Solo para ver bien que pasa).
} // Finaliza función loop()
```

Acá te dejo una mejor descripción de las partes



Otra opción es usar el optointerruptor como encoder, **lo normal es usar las interrupciones de Arduino**, lo que, si saben como usar, simplificará considerablemente el código, aunque eleva el nivel de conocimiento necesario para realizar el programa.

