CAPÍTULO 9: **SENSOR INFRARROJO**

Sensor infrarrojo

Un nuevo proyecto de experimentación en monitor serie, ésta vez con un sensor que esta conformado por 2 leds. Un led es emisor y el otro receptor, y al igual que el sensor ultrasónico, no podremos corroborar su correcto funcionamiento sino es por medio de una pantalla.

¡Si! Estás en lo cierto, la luz infrarroja si la podemos ver, pero a través de una cámara, ya que nuestro ojo no la percive. Podés comprobar el funcionamiento de emisor con la cámara de tu celular y las debidas conexiones (OJO! Si tenés iphone, posiblemente no podrás hacerlo, sus cámaras tienen filtro infrarrojo).

Consejo:

Vas a usar una gran cantidad de cables. Los sensores tienen que estar muy bien conectados, una pata floja hará que el experimento no funcione.

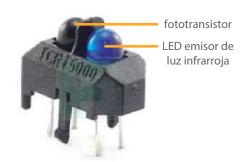


Sensor infrarrojo

El TCRT5000 es un sensor sencillo. Dispone de un LED emisor de luz infrarroja, y de un fototransistor que deja pasar más o menos corriente dependiendo de la luz que recibe.

Entre medio de ellos, hay una "barrera" plástica negra que evita que la luz llegue directamente desde el emisor al receptor.

El nivel de señal analógica que proporciona el sensor dependerá por tanto de la cantidad de luz recibida.



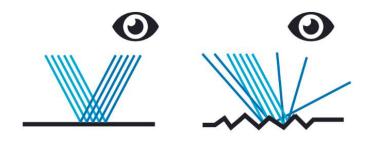
Factores que influyen en la medición del sensor

La cantidad de luz recibida puede depender de múltiples factores como:

- La distancia de reflexión, que es la distancia a la que se coloca el sensor con respecto a la superficie. La distancia óptima es de 2,5 mm. y como máximo unos 15 mm. dado que es un sensor de muy poco alcance.
- El color de la superficie, ya que hay colores que reflejan la luz mejor que otros.



- La cantidad de luz infrarroja en el ambiente. El sensor dispone de un filtro para eliminar los efectos de la luz ambiental y de día, pero su uso en entornos exteriores está totalmente desaconsejado.
- El tipo de superficie. En las superficies lisas la luz rebota de manera regular, mientras que en las porosas no, por lo que el valor detectado será menor.







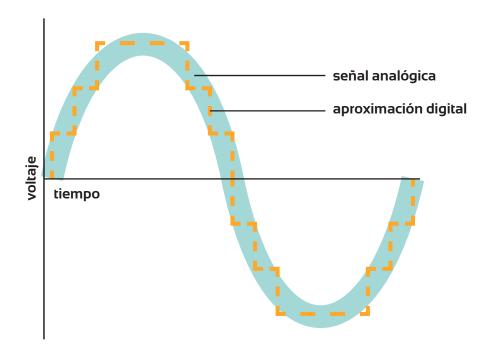
Usos adecuados para el sensor infrarrojo

Este tipo de sensores puede utilizarse para diversos propósitos, como:

- La detección de objetos (es decir, si están presentes o no) como un contacto de proximidad, ya que disponen de un corto alcance. El LED emite los infrarrojos, que en principio se pierden en el espacio, pero si se coloca algo delante del LED estos infrarrojos golpearán y rebotarán, llegando al fototransistor y cambiando la cantidad de corriente que éste deja pasar, permitiéndole detectar así la presencia de un cuerpo u otro objeto cercano.
- En robótica, su principal utilidad es en aplicaciones del tipo seguidor de líneas en la que el sensor devolverá un valor distinto en función del color que tenga delante. Por ejemplo, si el infrarrojo está sobre la línea de color negro el nivel de reflexión es muy bajo y el sensor devuelve un valor de señal elevado (lógica contraria). En cambio, si emite su señal sobre el suelo de color blanco el sensor apenas devuelve señal.

Señales eléctricas analógicas

Una **señal eléctrica analógica** es aquella en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. **Un microcontrolador** no tiene capacidad alguna para trabajar con señales analógicas, de modo que **necesita convertir las señales analógicas en señales digitales para poder trabajar con ellas**.



analogRead

Esta función sirve para realizar la lectura de los valores analógicos enviados desde los sensores a los pines del arduino. En el caso de un arduino Uno, el valor de O voltios analógico es expresado en digital como 0000000000 (0) y el valor de 5V analógico es expresado en digital como 1111111111 (1023). Por lo tanto, todo valor analógico intermedio es expresado con un valor entre O y 1023, es decir que suma 1 en binario cada 4,883 mV.





Arduino Uno tiene una resolución de 10 bits, es decir, puede devolver valores entre 0 y 1023.

Arduino Due tiene una resolución de 12 bits, es decir, puede devolver valores entre 0 y 4095.

En nuestro arduino, los únicos pines con la capacidad para leer las señales analógicas están ubicados a la izquierda de la placa arduino e identificados con valores que van desde el AO al A5. A diferencia de los digitales, estos pines por defecto son INPUT por lo que no es necesario usar el pinMode si vamos a usarlos como entrada. Cabe aclarar que, aunque se los llama pines analógicos, también tienen todas las funcionalidades de los pines digitales. Por lo tanto, si necesitamos más pines digitales también podemos usarlos.

Su sintaxis correcta es:

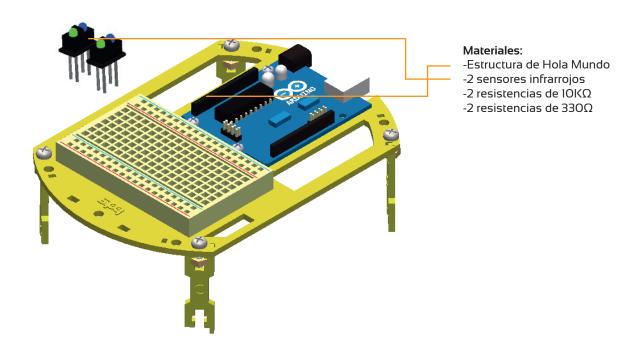
analogRead(pin);

Ejemplo:

valor = analogRead(2); // hace que la variable 'valor' sea igual al valor leído en el pin A2.

Ensamble

A continuación veremos la estructura y componentes necesarios para hacer nuestra primer prueba de motores DC.

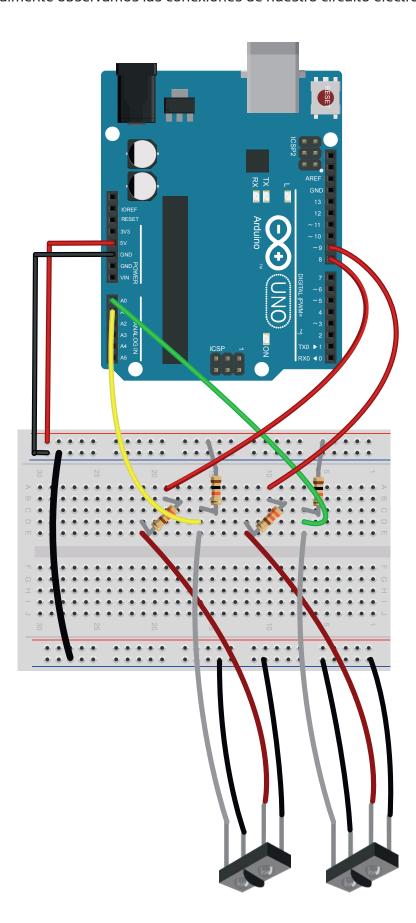






Circuito electrónico

Finalmente observamos las conexiones de nuestro circuito electrónico.

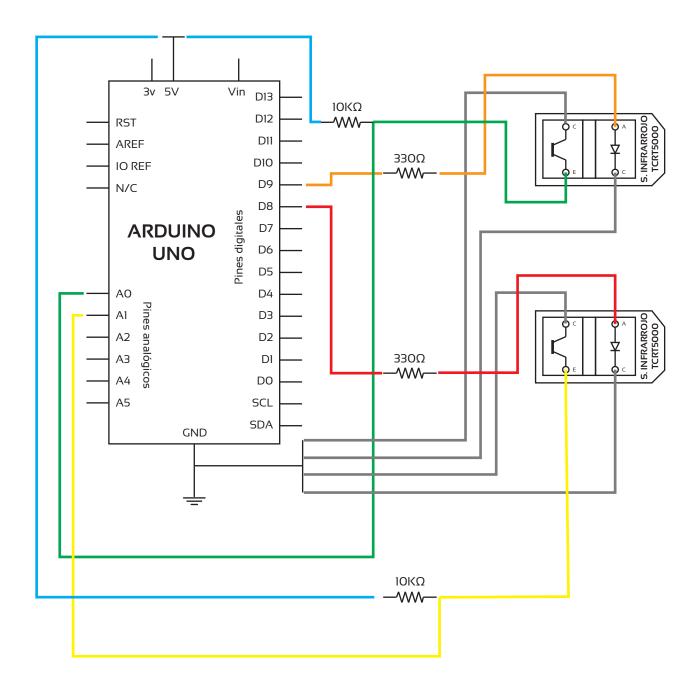


Componentes electrónicos:

- -2 sensores infrarrojos
- -2 resistencias de $10 \text{K}\Omega$
- -2 resistencias de 330Ω
- -7 cables M-M
- -8 cables M-H



Circuito electrónico en símbolos





Código de programación

Desarrollaremos el código de programación para calibrar nuestros sensores infrarrojos, observando los valores registrados por ellos a través de nuestro monitor serie.

```
1 const int irlzquierda = 9;
                                                 //Definimos las variables que van a vincular salidas
 2
    const int irDerecha = 8;
                                                 y entradas del arduino: las salidas a encender los
 3
    const int ledTest = 13;
                                                 led infrarrojos en el pin 8 y 9. La información la
 4
                                                 vamos a leer con analogRead en las entradas AO y
 5 const int trlzquierda = AO;
                                                 Al. Dependiendo de la reflectividad va a ser el valor
    const int trDerecha = A1;
 7
 8 int refDer = 0;
 9
    int reflzq = 0;
10
 11 void setup() {
                                                 //Con Serial.begin(9600) inicializamos / habilita-
12
    Serial.begin(9600);
                                                 mos la entrada y salida de datos por el puerto serie.
13
     pinMode(irlzquierda, OUTPUT);
14
     pinMode(irDerecha, OUTPUT);
15
     pinMode(ledTest, OUTPUT);
16
17
     digitalWrite(irlzquierda, LOW);
18
     digitalWrite(irDerecha, LOW);
19
     digitalWrite(ledTest, LOW);
20
21
     digitalWrite(ledTest, HIGH);
                                                 //Saludo Inicial.
22
     delay(500);
23
     digitalWrite(ledTest, LOW);
24
     delay(500);
25
     digitalWrite(ledTest, HIGH);
26
     delay(500);
     digitalWrite(ledTest, LOW);
27
28
     delay(500);
     digitalWrite(ledTest, HIGH);
29
30
     delay(500);
     digitalWrite(ledTest, LOW);
31
32
     delay(500);
33 }
34
35
36
    void loop() {
37
38
     digitalWrite(irlzquierda, HIGH);
                                                 //Encendemos el led izquierdo y posteriormente
39
                                                 leemos la salida del fototransistor.
40
41
     reflzq = analogRead(trlzquierda);
                                                 //analogRead Lee el valor del pin analógico especi-
42
     delay(100);
                                                 ficado. mapeará tensiones de entrada entre O y 5
     digitalWrite(irlzquierda, LOW);
43
                                                 voltios en valores enteros entre O y 1023
44
     delay(100);
45
46
     digitalWrite(irDerecha, HIGH);
                                                 //Una vez leido apagamos el derecho para que no
47
     refDer = analogRead(trDerecha);
                                                 interfiera con la medición del lado derecho.
48
     delay(100);
     digitalWrite(irDerecha, LOW);
49
50
51
     Serial.print(reflzq);
                                                 //Enviamos al puerto serie la información recibida
52
     Serial.print(",");
53
     Serial.println(refDer);
54
55
     delay(500);
56 }
```

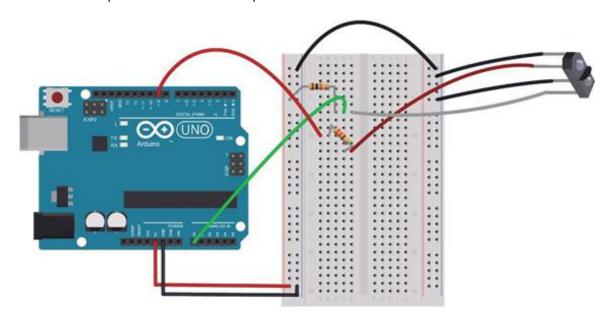
MATERIAL COMPLEMENTARIO



Testing de componentes

Sensor infrarrojo

Para probar este sensor es necesario armar un circuito sencillo, como el que se muestra a continuación. El proceso es similar al que vimos con el sensor ultrasónico.



Una vez listo el ensamble, tendrás que conectar el arduino a la PC, copiar el código de prueba que aparece en la página siguiente en el IDE, ejecutarlo y observar los resultados en el monitor serie.

Código para probar el sensor infrarrojo

```
const int ledTest = 13;
const int irlzquierda = 9;
const int trlzquierda = AO;
int reflzq = 0;
void setup() {
Serial.begin(9600);
 pinMode(irlzquierda, OUTPUT);
 pinMode(ledTest, OUTPUT);
 digitalWrite(irlzquierda, LOW);
 digitalWrite(ledTest, LOW);
}
void loop() {
 digitalWrite(irlzquierda, HIGH);
 reflzq = analogRead(trlzquierda);
 delay(100);
 digitalWrite(irlzquierda, LOW);
 delay(100);
 Serial.println(reflzq);
 delay(500);
```



ACTIVIDADES Y EJERCITACIÓN



Cap. 9: Actividades complementarias

En base a los contenidos vistos en el "Capítulo 9: Sensor Infrarrojo" realiza las siguientes actividades complementarias.

Actividad 1

Desafío práctico: identificar colores.

- **a.** Modifica el programa visto en esta unidad para que el sensor indique en el monitor serie si el objeto que tiene delante es blanco o negro.
- **b.** Si te atreves a probar algo un poco más complejo, usa la estructura **if / else if** para identificar distintos tonos de grises.

Actividad 1.2

Agrega dos leds al circuito, de manera tal que si el objeto mostrado es negro encienda un led rojo, mientras que si el objeto es blanco encienda un led también blanco.

Actividad 1.3

Desarrolla un circuito y la programación correspondiente para que, incorporando un led RGB, éste genere distintos colores según los colores del objeto que tenga al frente.

Actividad 1.4

Usa un LED blanco que varíe su intensidad al variar el tono del objeto que tiene en frente.

Por ejemplo:

- Si el objeto es negro, que el led esté apagado,
- Si es gris, que el led aumente su intensidad según qué tan claro u oscuro sea,
- Si es blanco, que el led esté encendido al 100%.

<u>Tip:</u> Para lograrlo se puede usar la función map.





Cap. 9: Respuestas

Actividad 1

Desafío práctico: identificar colores.

a. Para que el sensor indique en el monitor serie si el objeto es blanco o negro.

```
const int irlzquierda = 9;
const int irDerecha = 8;
const int trlzquierda = AO;
const int trDerecha = A1;
//Datos reflectividad guardada
int refDer = 0;
int reflzq = 0;
void setup() {
 Serial.begin(9600);
 pinMode(irlzquierda, OUTPUT);
 pinMode(irDerecha, OUTPUT);
 digitalWrite(irlzquierda, LOW);
 digitalWrite(irDerecha, LOW);
}
void loop() {
 digitalWrite(irlzquierda, HIGH);
 reflzq = analogRead(trlzquierda);
 delay(100);
 digitalWrite(irlzquierda, LOW);
 delay(100);
 digitalWrite(irDerecha, HIGH);
 refDer = analogRead(trDerecha);
 delay(100);
 digitalWrite(irDerecha, LOW);
/* Primero buscamos el umbral con Serial.Print y, una vez que sabemos que por debajo de 700 es
blanco y mayor a 1010 es negro, creamos los IF necesarios. Hacemos dos IF separados para que los
valores intermedios entre 700 y 1010 no nos den falsos positivos, es decir, que diga "Negro" cuando es
un gris. Este tip te servirá para el punto b de la consigna, donde se desafía a identificar distintos tonos
de grises */
 if (reflzq < 700 && refDer < 700) Serial.println("Blanco");
 if (reflzq >1010 && refDer >1010) Serial.println("Negro");
 delay(500);
```

b. Usa la estructura **if / else if** para identificar distintos tonos de grises. *Ver anotaciones en el código de arriba.*





Actividad 1.2

Utilizando de base el código visto en la **Actividad 1**, añade las constantes para el led blanco y el led rojo. Incorpora dentro de cada uno de los IF el código necesario para que, si la especificación se cumple, encienda el led correspondiente.

Esta actividad combina los contenidos vistos en "Luces para mi robot" e IF / IF... ELSE.

Actividad 1.3

Utilizando de base el código visto en la **Actividad 1**, añade las constantes del led RGB. Incorpora dentro de cada uno de los IF el código necesario para que, si la especificación se cumple, ilumine el led de determinado color.

Esta actividad combina los contenidos vistos en "Led RGB" e IF / IF... ELSE.

Actividad 1.4

Utilizando de base el código visto en la **Actividad 1**, añade la constante del led blanco. Incorpora la función map para controlar la intensidad del led según el color que detecte el sensor infrarrojo.

Si precisas repasar cómo se usa la **función map**, revisa **"Capítulo 7: Calibración y Uso"**.

