

PRÁCTICA 10: INTERRUPTOR CON SENSOR DE INFRARROJOS

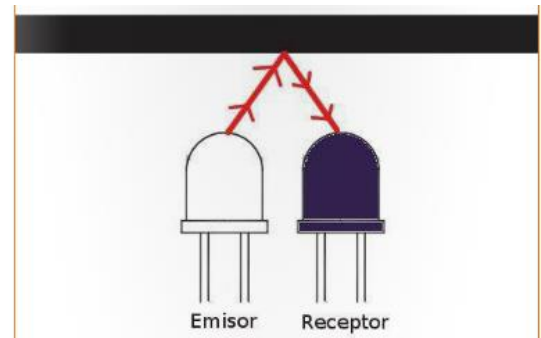
El objetivo de esta práctica es conseguir detectar un objeto programando el sensor óptico de infrarrojos en S4A.

El sensor óptico de infrarrojos (modelo TRCT5000) posee 3 terminales. Dos de ellos se conectarán a tierra (GND) Y 5V en la placa arduino y el tercero, llamado OUT, se conectará a una entrada analógica, en nuestro caso A1. Para comprobar el funcionamiento del sensor usaremos un LED rojo conectado al pin 13, que se iluminará al detectar el sensor un objeto.



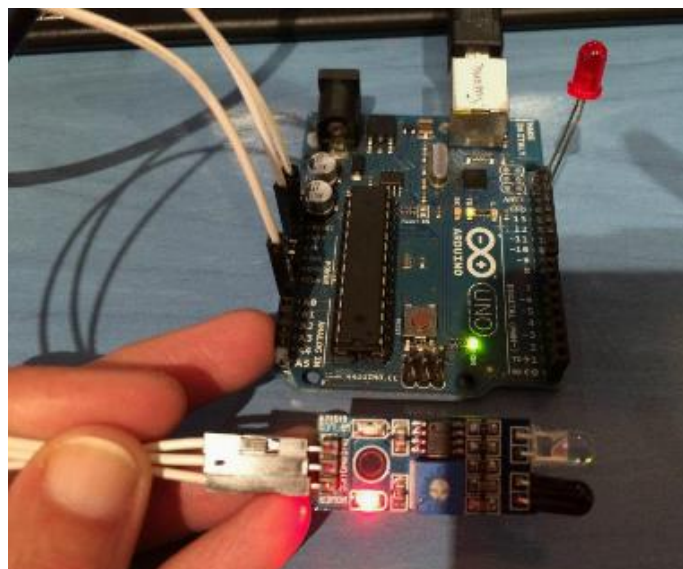
Sensor de IR

Este sensor de infrarrojos consta de un diodo emisor de luz infrarroja y un fototransistor como receptor de la luz infrarroja reflejada. El sensor funciona por reflexión, de modo que el emisor emite luz infrarroja y cuando algún objeto se interpone en dicho rayo de luz, se refleja y llega al fototransistor, dando éste la orden al LED para que se encienda.



El modelo de sensor de infrarrojos que usaremos en esta práctica detecta objetos a distancias desde 2 cm a 60 cm. Esta distancia de detección puede ajustarse (proceso de **calibración**) mediante el potenciómetro que dispone: ajustar el potenciómetro en sentido horario aumenta la distancia de detección, y ajustarlo en sentido antihorario la reduce.

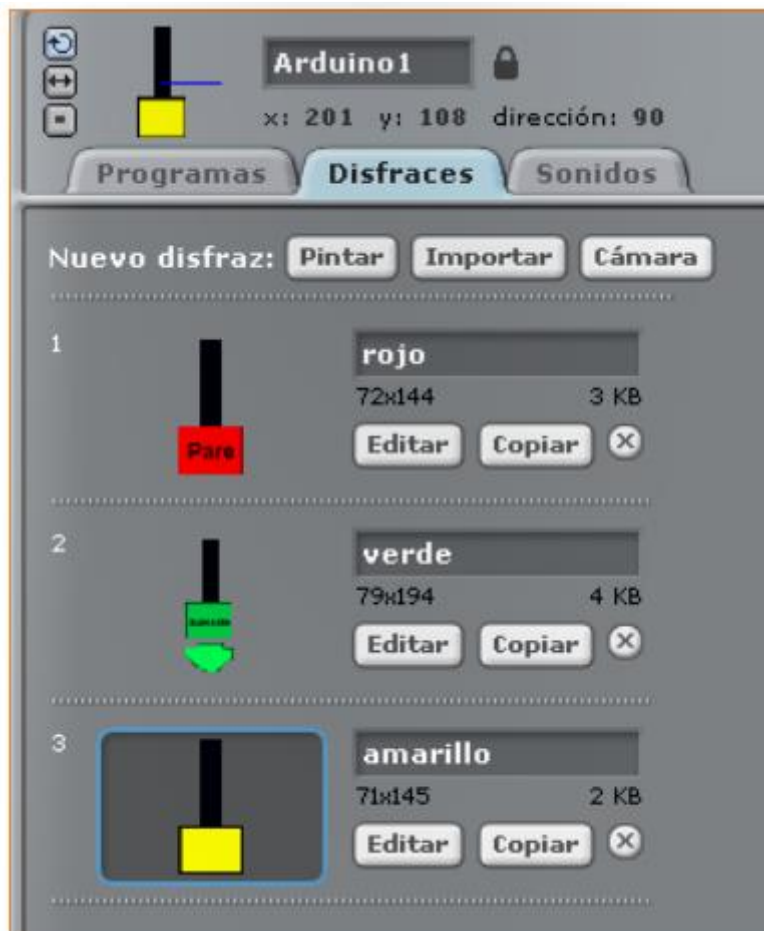
La conexión del sensor a la placa Arduino se observa en la siguiente imagen, en la que como ya se ha dicho, se observa el LED rojo conectado al pin 13 y el pin OUT del sensor al pin analógico A1, mientras que los otros cables del sensor van a 5V y GND de la placa.



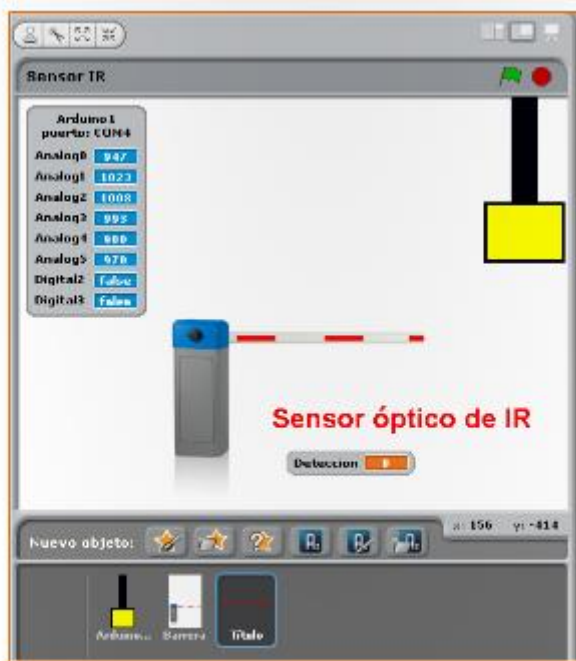
Objetos en el entorno S4A

El programa que se crea en el entorno S4A simula la detección de un objeto por proximidad. Para ello utilizo 3 objetos: el objeto arduino, el objeto barrera y el objeto título de la práctica. Este último no presenta ningún interés en cuanto a la programación.

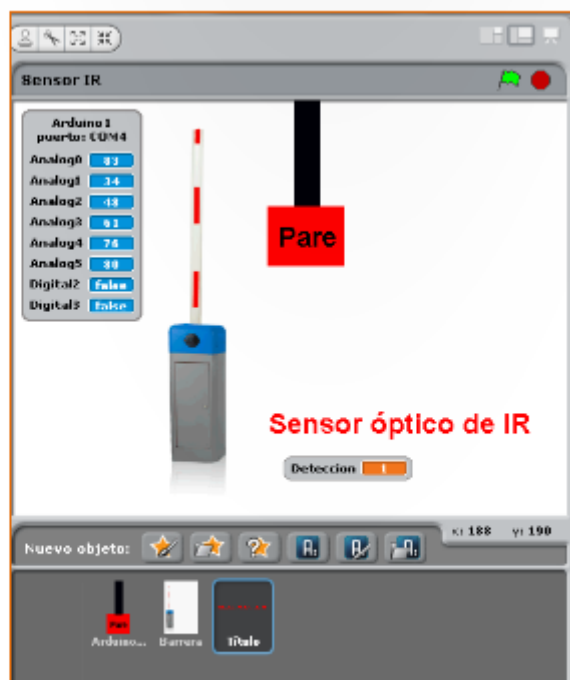
Objeto arduino presenta tres disfraces (amarillo, rojo y verde) y será programado para encender el LED rojo de la placa arduino cuando el sensor detecte la presencia de un objeto delante del sensor de IR. El disfraz "amarillo" es el que se mostrará cuando el sensor no detecte un objeto cercano y por eso la variable "Detección" toma el valor cero. Los otros dos disfraces (rojo y verde) se mostrarán al detectar el objeto y en ellos, la variable "Detección" toma el valor 1.



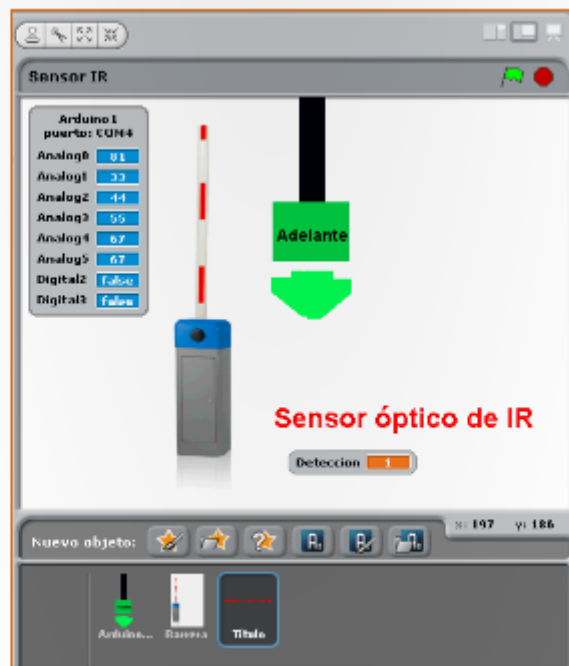
En la simulación se ha pretendido crear una barrera de modo que esta suba y baje. Por lo tanto, mientras sube y mientras baja, el objeto no puede pasar y eso se refleja en el disfraz "rojo" (pare) del objeto arduino. En el momento en que la barrera está completamente subida, ésta se quedará quieta por unos segundos y ahí es cuando el disfraz del objeto arduino cambia a "verde" (adelante).



Escenario en "Detección=0": disfraz amarillo.



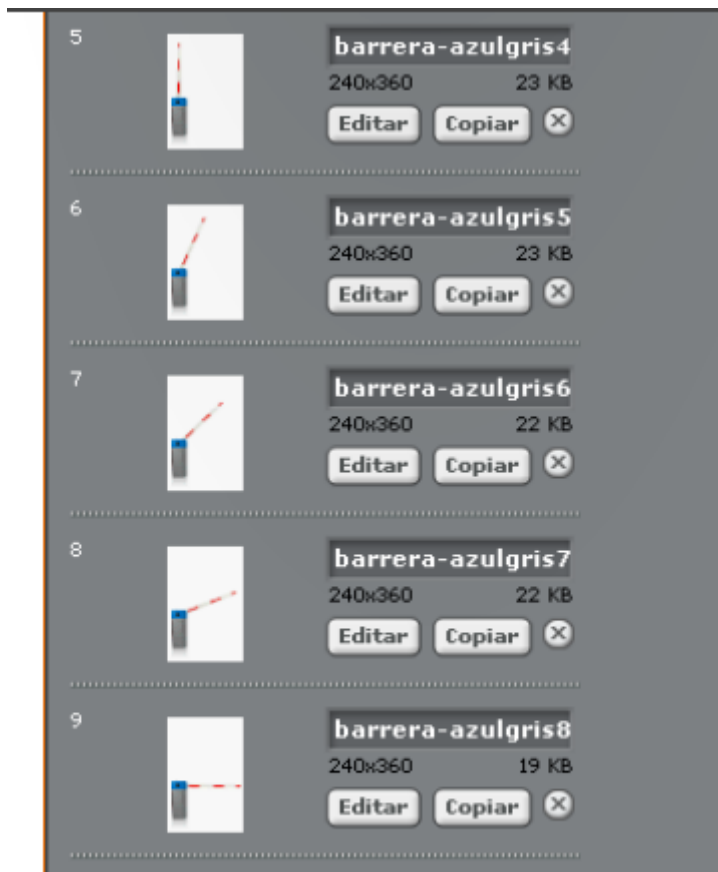
Escenario en "Detección=1": disfraz rojo subida de barrera.



Escenario en "Detección=1": disfraz verde subida completada.

El objeto barrera presenta 9 disfraces que se programan de forma cíclica llamando a los respectivos disfraces del objeto arduino (amarillo, rojo y verde).





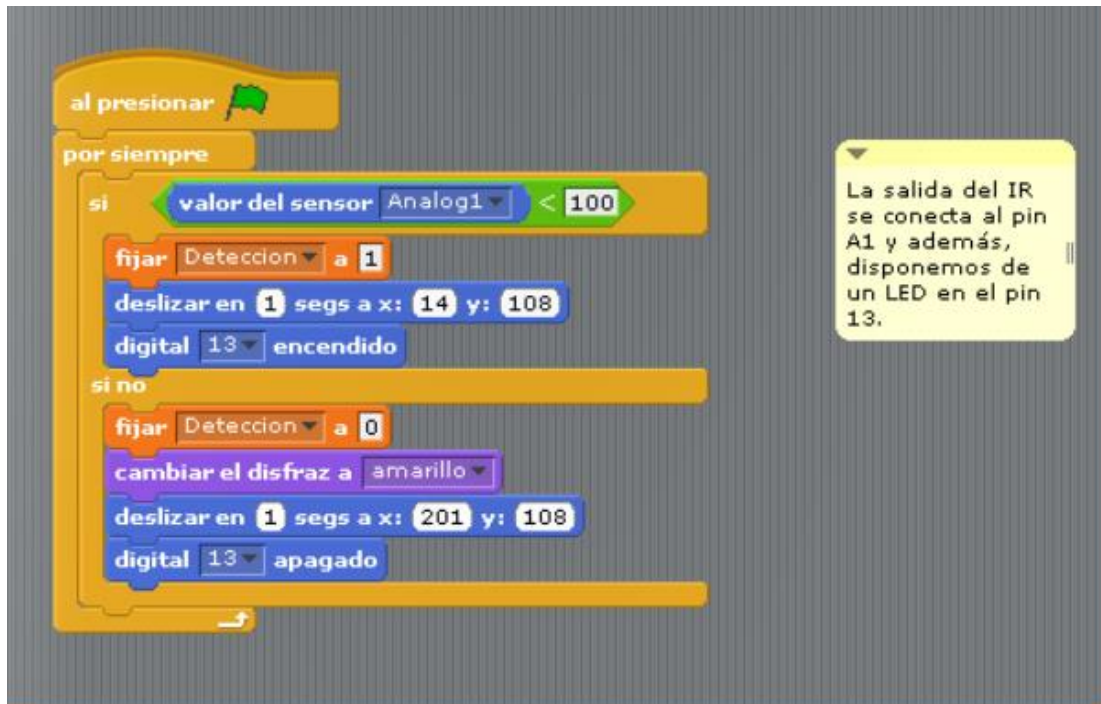
Disfraces del objeto Barrera.

El objeto título simplemente se ha creado para mostrar en una posición determinada del escenario (99, -101) el título de la práctica.

[Programación en el entorno S4A.](#)

El sensor de IR se programa de modo que, al detectar la proximidad de un objeto, nos avise encendiendo un LED Rojo del pin digital 13 de la placa arduino. El valor que mide el sensor en su pin OUT se refleja en la entrada analógica 1 de la placa. El sensor se ha ajustado inicialmente a una cierta distancia que se asocia con el número 100 o menor de la entrada analógica A1 (para un número superior, el objeto se considera alejado y no lo detecta, mostrándose el diodo rojo apagado). Para facilitar la comprensión del programa se ha creado una variable que llamo "Detección" y que tomará 2 valores: 1 en el caso de que detecte al objeto en cercanía y 0 si el objeto no es detectado.

El programa se muestra en la siguiente imagen:



La salida del IR se conecta al pin A1 y además, disponemos de un LED en el pin 13.

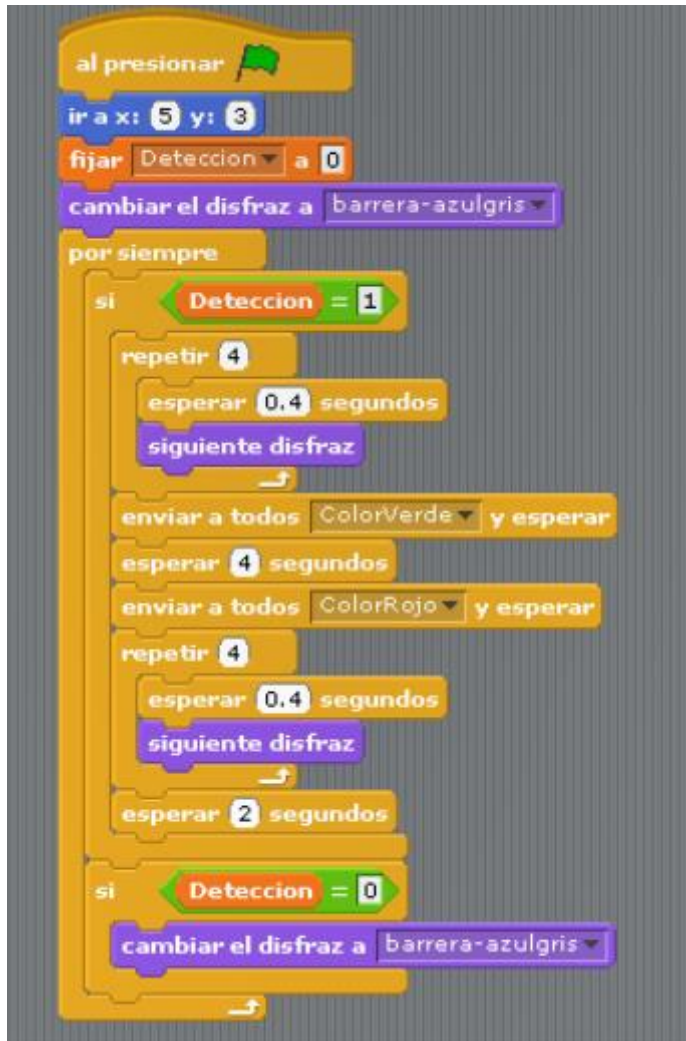
Programación del sensor IR en el objeto arduino.

Inicialmente el objeto arduino se muestra con el disfraz amarillo en una posición determinada del escenario. Sólo resta programar la aparición de los disfraces "rojo" y "verde" a partir del objeto "barrera":



Programación de los disfraces del objeto arduino.

En el objeto "Barrera" sólo debe programarse la secuencia de aparición de sus disfraces, así como el momento en que se activan. Inicialmente, cuando el objeto arduino presenta el disfraz "amarillo" la barrera debe estar en el disfraz horizontal (barrera-azulgris) y la variable "Detección" toma el valor cero, pero al detectar el sensor el objeto (Detección=1), los disfraces de la barrera deben ir pasando secuencialmente (4 disfraces implica un bucle de repetición de 4 unidades) hasta lograr mostrar la barrera subida. Allí, el disfraz del objeto arduino debe cambiar a "verde" durante unos segundos y ponerse en "rojo" cuando la barrera vuelva otra vez a bajar (otros 4 disfraces y por este motivo usamos un bucle de 4 repeticiones de cambio de disfraz). El programa que refleja esta secuencia es el siguiente:



Finalmente, el objeto título se programa para que aparezca un texto en una determinada posición del escenario. En concreto, en (99, -101).

En este **otro ejemplo** vamos a utilizar un emisor y un receptor infrarrojos para construir un interruptor que se acciona con luz IR. Este interruptor hará que se encienda un LED cuando algún objeto se interponga entre el emisor y el receptor de infrarrojos.

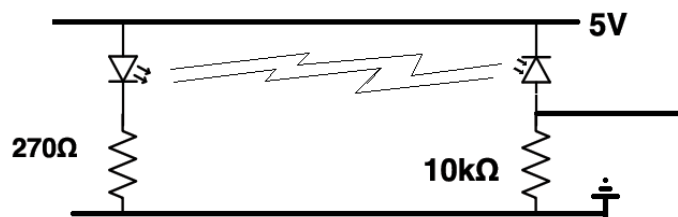
En esta práctica vamos a utilizar un sensor de infrarrojos con emisor y receptor separados, que tienen un aspecto muy parecido a los leds, el de color oscuro es el emisor y el transparente es el receptor.

Conexión en la protoboard:

La polaridad del emisor es similar a la de los led, se conecta con una resistencia pequeña (270Ω) y con la pata más larga conectada a los 5V y la más corta hacia tierra.

En el receptor la polaridad es la contraria, con la pata más larga hacia tierra y la más corta hacia los 5V. En el circuito el receptor se comportará como un interruptor, por lo tanto colocaremos una resistencia grande ($10k\Omega$) entre la pata larga y tierra. El punto de conexión a la placa arduino será la unión entre el receptor IR y la resistencia de $10k\Omega$ que ira conectado a la entrada Analog1

El circuito lo completaremos conectando un led en la salida 13 (sin resistencia), que lo encenderemos cuando se corta la comunicación entre el emisor y el receptor IR.

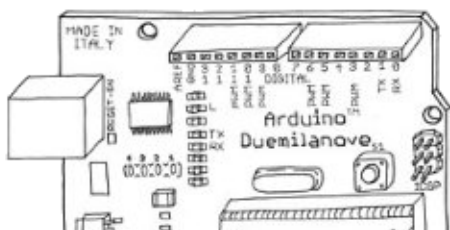


El código sería el siguiente:

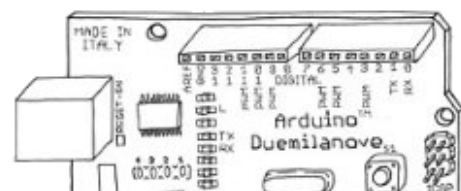


Para establecer el valor que se comparará con el recibido en la entrada Analog 1 se ha utilizado 800, por encima de este valor la condición será cierta y por debajo será falsa. Este valor es el que define si se enciende el led o no. En [el vídeo](#) se ve cómo al interrumpir la comunicación se enciende el led y al quitar el obstáculo se apaga.

Arduino1 puerto: fd131	
Analog1	1000
Analog2	980
Analog3	959
Analog4	940
Analog5	919
Analog6	901
Digital1	false
Digital2	false



Arduino1 puerto: fd131	
Analog1	256
Analog2	253
Analog3	251
Analog4	251
Analog5	249
Analog6	252
Digital1	false
Digital2	false



Como podemos ver en las imágenes el valor de la entrada cuando la comunicación está interrumpida (la luz infrarroja no llega al receptor) es de 256 y cuando hay conexión es de 1000. El valor de 256 está producido por la luz ambiental que llega al receptor. Si queremos valores más extremos (en la parte baja) podemos proteger el receptor con una capucha para disminuir la luz ambiental que le llega.