

PRACTICAS

INTERMEDIO

Prácticas ya con sensores como el LDR, el sensor de distancia Ultrasonidos

Salidas con actuadores como el servo

Lectura de valores de los sensores por el puerto serie

- Interruptor crepuscular
- Barrera por ultrasonidos
- Comunicaciones
- Lectura puerto serie ultrasonidos

Interruptor crepuscular

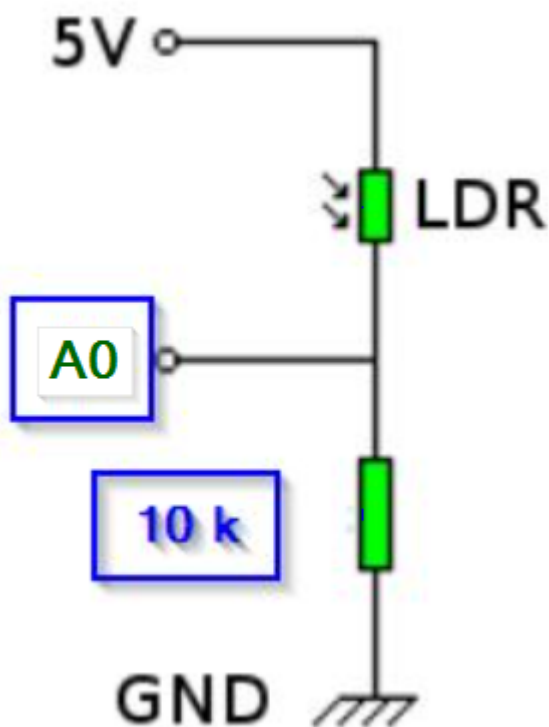
Objetivo

Realizar un interruptor crepuscular, es decir, cuando se oscurece, que se encienda la luz y que se apague cuando la luminosidad se recupere.

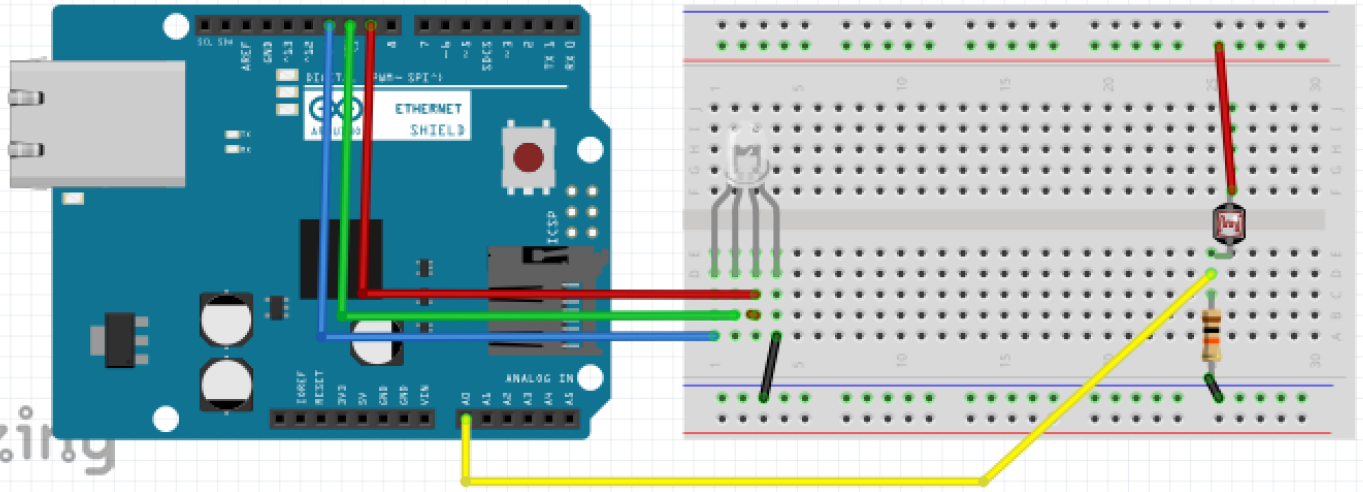
https://www.youtube.com/embed/1d_siWmJwiM

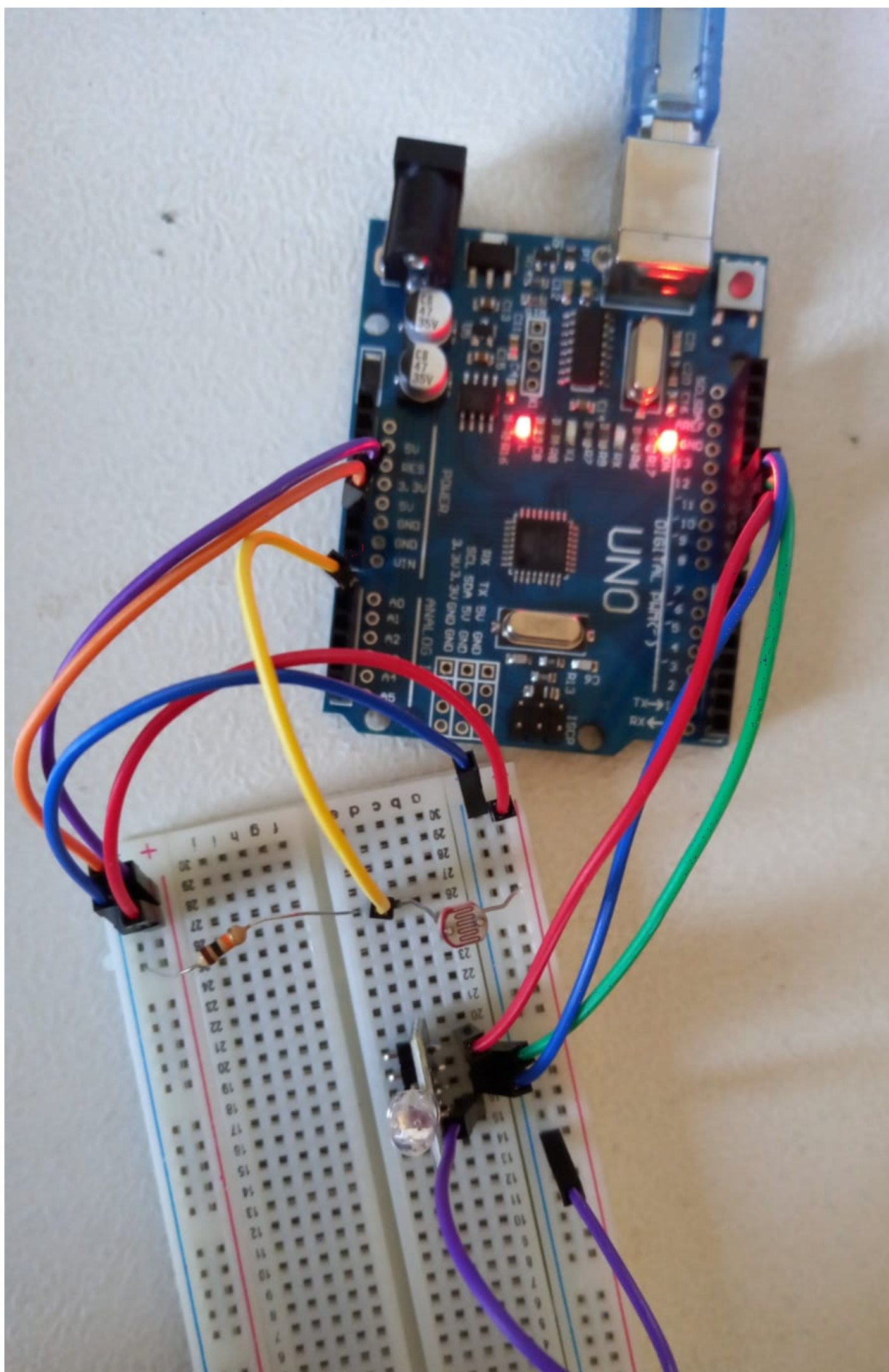
Cableado

Utilizaremos el LDR con un divisor de tensión de tal manera que al modificar el LDR su valor, la tensión que han entre las dos resistencias se modifica. El punto intermedio que lee la tensión lo conectaremos a una entrada analógica, al A0 por ejemplo.



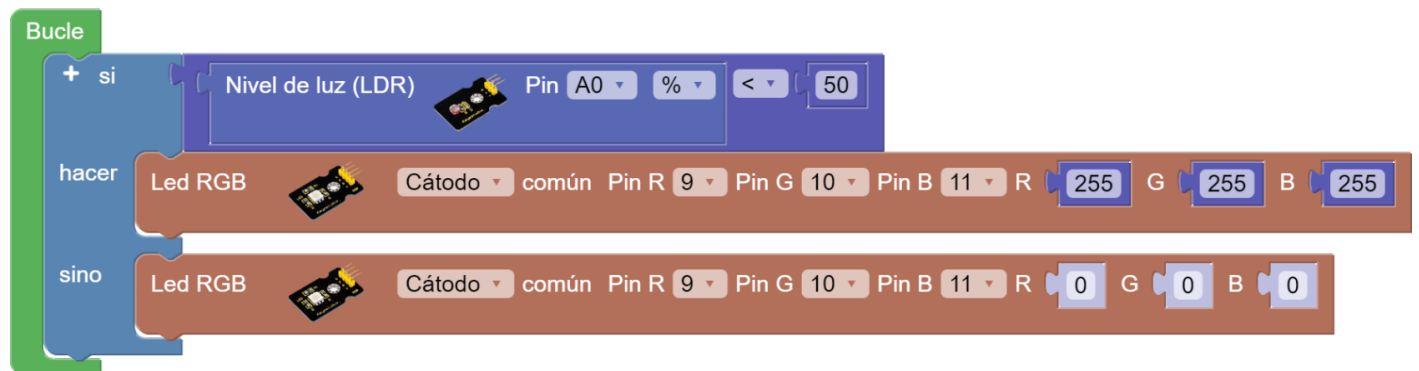
zing





Programa

<http://www.arduinoblocks.com/web/project/780612>



Barrera por ultrasonidos

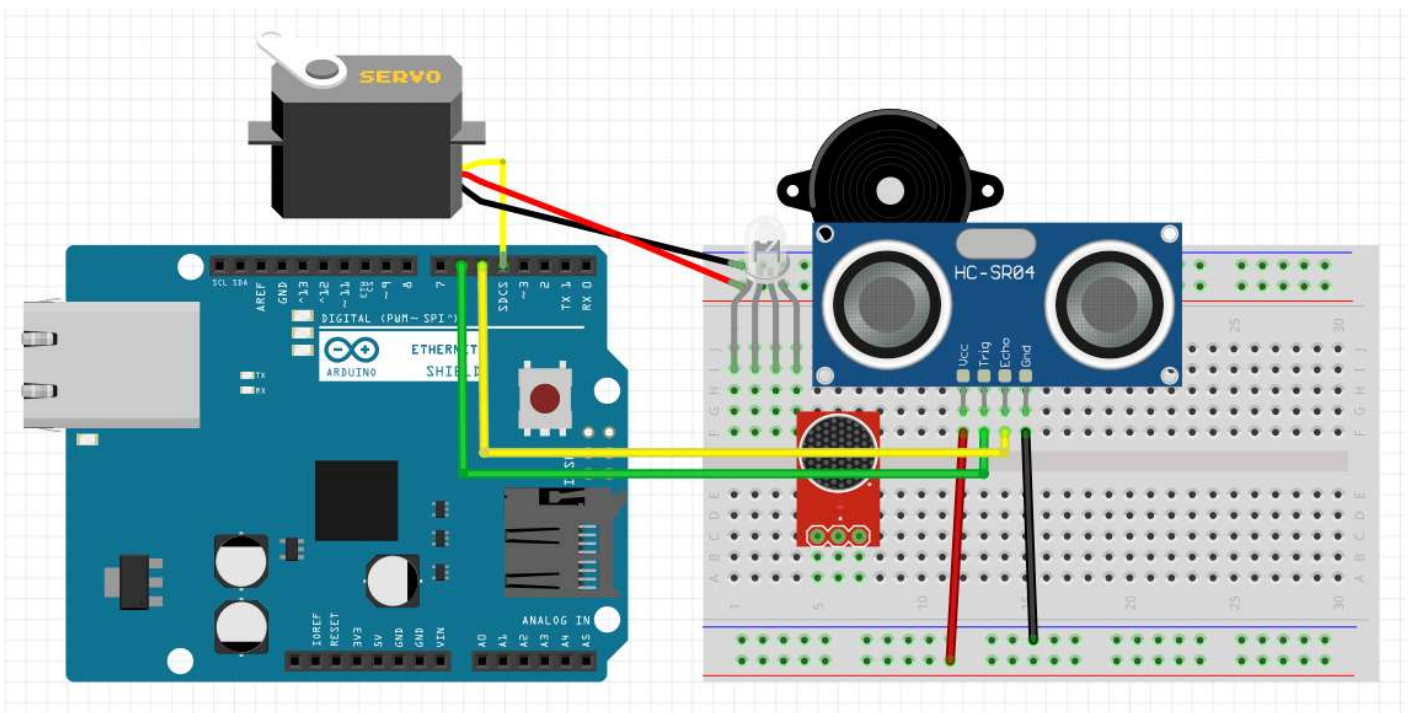
Objetivo

Crear un programa que al detectar un objeto a menos de 10cm suba una "barrera" simulada con un servo y un trozo de papel (en el vídeo un *"postit"*)

<https://www.youtube.com/embed/L5nhbA--yBk>

Cableado

ATENCIÓN Se han omitido las conexiones del led RGB y las del zumbador



Programa

El programa lo tienes aquí <http://www.arduinoblocks.com/web/project/770872>

Inicializar

Bucle

+ si

Distancia (cm)

[Trigger] 6

[Echo] 5

< 10

hacer

Led RGB

Cátodo

común

Pin R 9

Pin G 10

Pin B 11

Color

Zumbador

Pin 3

Ms

100

Hz

Tono (Hz)

DO

Servo

Pin 4

Grados

Ángulo 90°

Retardo (ms)

0

sino

Servo

Pin 4

Grados

Ángulo 0°

Retardo (ms)

0

Puedes mejorar el programa añadiendo un retardo en la bajada de la barrera, simulando el caso real, que cuando el coche pasa, hay un poco de retraso en el cierre de la barrera

También puedes añadir al programa que si se pulsa el pulsador táctil, que se suba la barrera, como pulsador de seguridad.

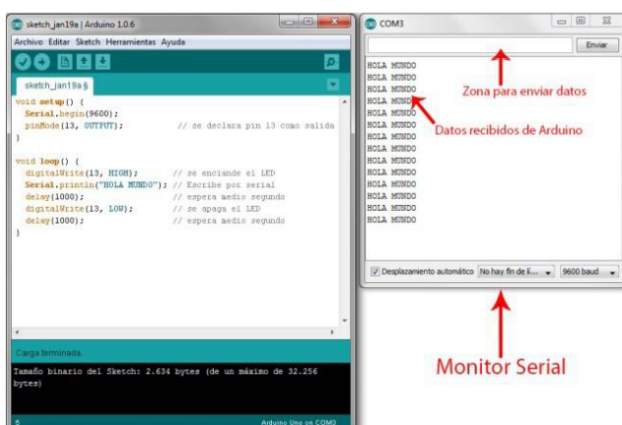
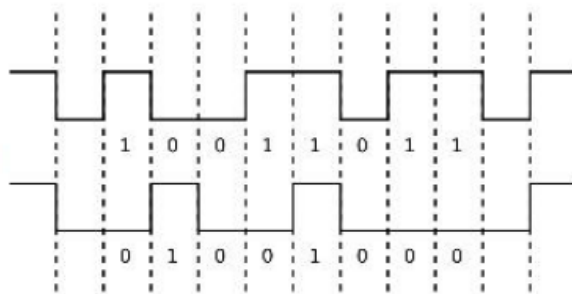
Comunicaciones

Arduino permite comunicarse con periféricos o con el PC de diferentes formas:

- **Puerto serie (COM):** Permite una conexión serie (tipo RS232) a través del puerto USB. Utilizada principalmente para programar la placa Arduino desde el ordenador y para comunicar con aplicaciones en un ordenador (consola serie). Con una aplicación de terminal o consola serie podemos visualizar los datos recibidos desde Arduino y enviar datos al Arduino fácilmente.

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Para utilizar este puerto hay que determinar antes a qué velocidad se van a transmitir los datos para que todo se sincronice. A la hora de programar en el apartado de *Inicio* **se determina la velocidad en baudios bits por segundo** normalmente 9600



Consola serie IDE de Arduino

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

ArduinoBlocks :: Consola serie

Baudrate: 9600

Lector RFID:
77bb203b

Consola serie ArduinoBlocks

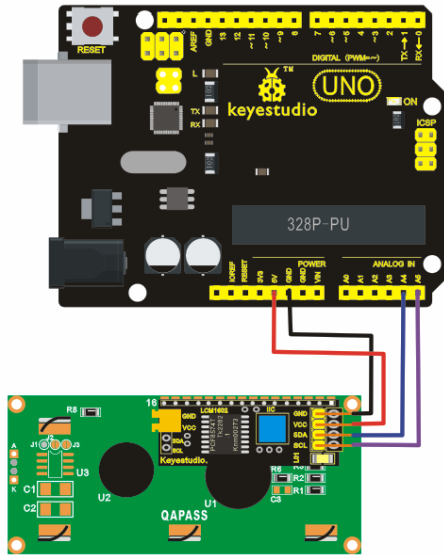
La conexión serie de Arduino está accesible en los pines 0 (RX) y 1 (TX), internamente estos pines están conectados al chip que gestiona el USB, en caso de usar la comunicación serie con otros dispositivos conectados a los pines 0 y 1 se deberá desconectar el USB para evitar que se mezclen los datos y no funcione la comunicación.

- **I2C:** Es otro sistema de bus de comunicaciones en serie. Permite crear una pequeña red de dispositivos (pensado principalmente para interconectar otros chips en la propia placa o cerca). La comunicación I2C sólo utiliza 2 cables.
 - Para el Arduino UNO utiliza SDA=A4 SCL=A5
 - Para el Arduino Nano utiliza SDA=A4 SCL=A5
 - Para el Arduino Mega utiliza SDA=20 SCL=21
 - Para I2C Keystudio UNO utiliza : SDA=A4 SCL=A5
- Existe una tercera forma de comunicación, la **SPI** (Serial Peripheral Interface), pero no la veremos en este manual.

(adaptado de José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

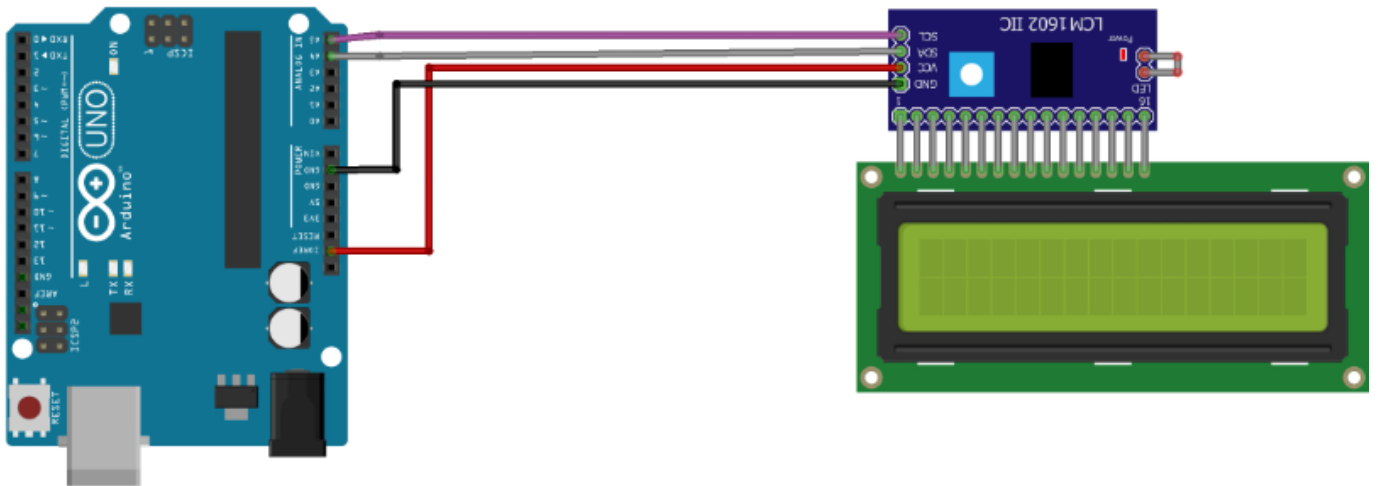
El **Puerto serie COM** lo utilizaremos como comunicaciones entre ordenador y las placas, aunque aquí puedes ver un ejemplo sencillo de comunicación entre dos Arduinos utilizando los pines 0-RX y 1-TX

El **I2C** se utiliza principalmente en la conexión con la pantalla I2C:



Ejemplo de conexión de LCD en el I2C con
Keystudio UNO

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)



Ejemplo de conexión de LCD con **ARDUINO UNO** para ver cómo se programa con código [aquí](#)

Lectura puerto serie ultrasonidos

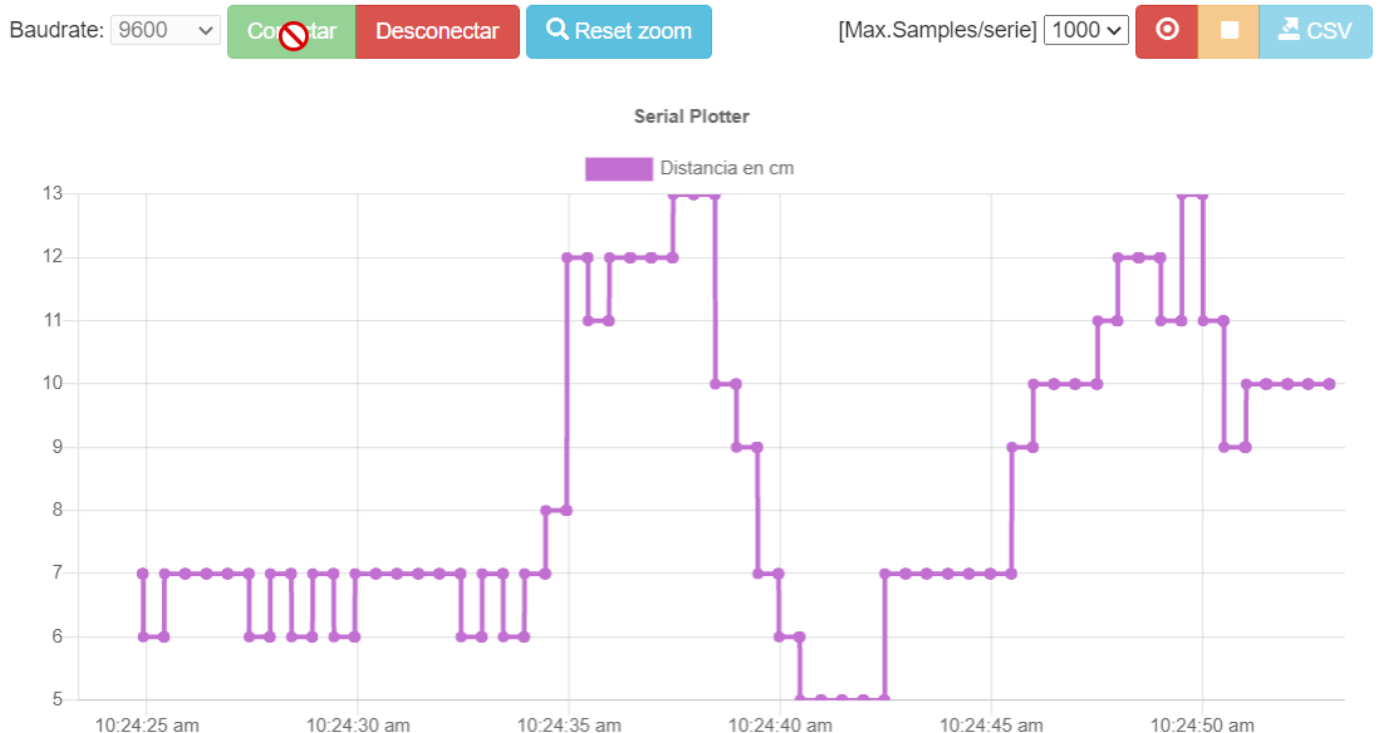
Objetivo

Vamos a visualizar por el puerto serie el valor del sensor de Ultrasonidos con el objetivo de aprender sobre las comunicaciones del Arduino y el ordenador y sus posibilidades de recogida de datos

La gráfica que obtendremos al pinchar en el serial plotter de arriba a la derecha será algo así, moviendo un obstáculo delante del sensor :

ArduinoBlocks :: Serial plotter + Datalogger

x



Arriba a la derecha tiene las opciones de exportar los datos a una hoja de cálculo. También se puede visualizar de forma textual en la **consola serie**

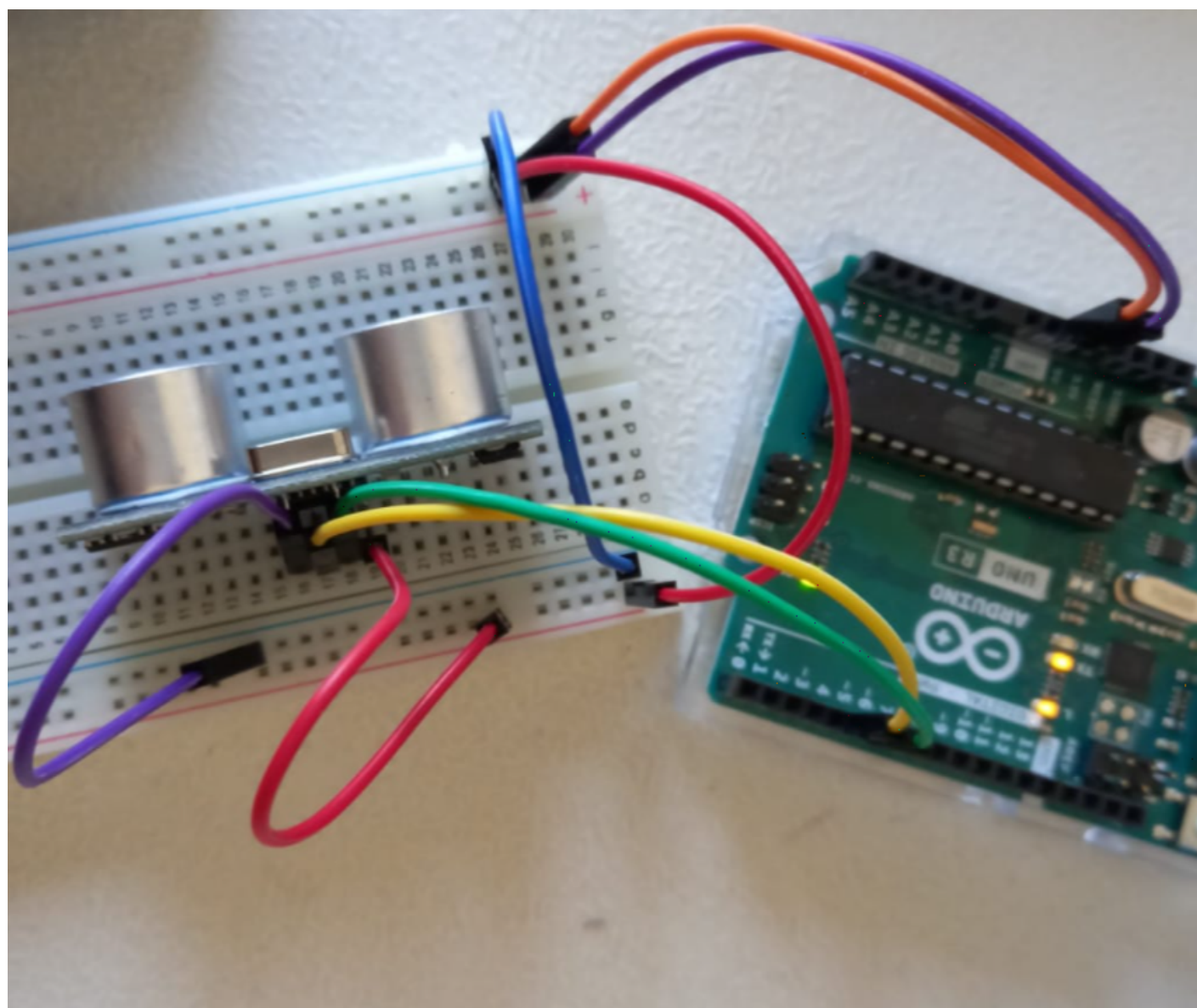
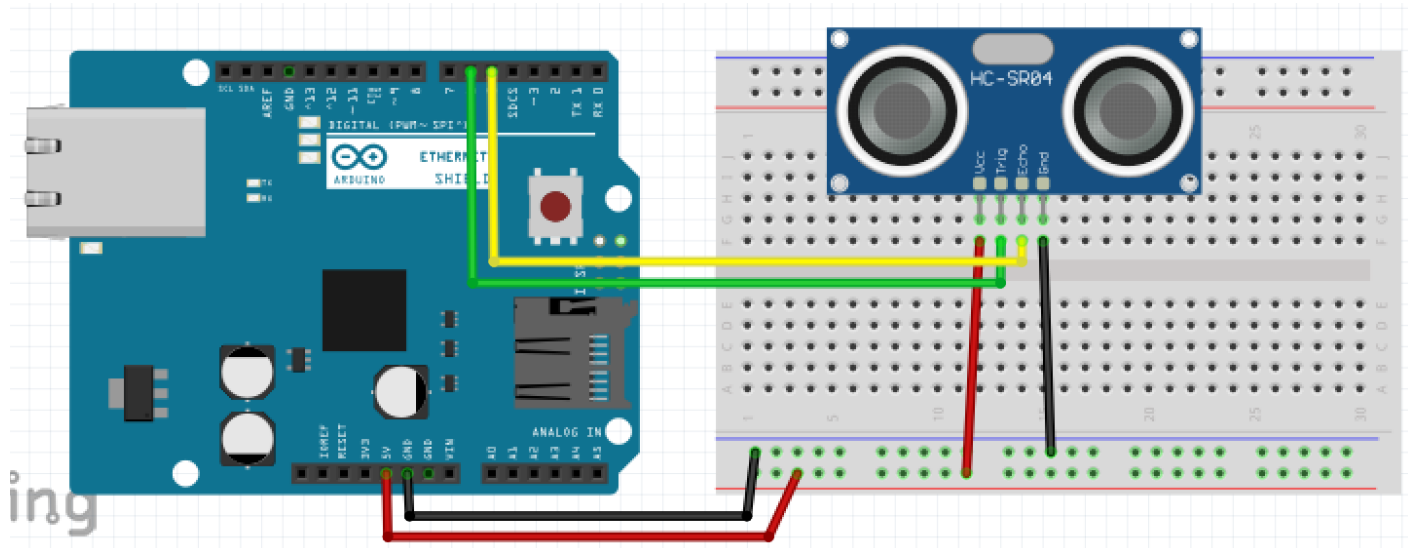
ArduinoBlocks :: Consola serie

Baudrate: 

Distancia en cm=10.00
Distancia en cm=10.00
Distancia en cm=10.00
Distancia en cm=10.00
Distancia en cm=10.00
Distancia en cm=12.00
Distancia en cm=15.00
Distancia en cm=17.00
Distancia en cm=21.00
Distancia en cm=22.00
Distancia en cm=14.00
Distancia en cm=8.00
Distancia en cm=7.00
Distancia en cm=6.00
Distancia en cm=6.00
Distancia en cm=6.00
Distancia en cm=6.00

Cableado

Simplemente conectar el sensor que se quiere examinar. Trigger en D6 y Echo e D5 por ejemplo:



Programa


<http://www.arduinoblocks.com/web/project/780686>

Inicializar

>_ Iniciar Baudios 9600 ▾

Bucle

>_  Plotter “ Distancia en cm ” Valor  Distancia (cm) [Trigger] 6 ▾ [Echo] 5 ▾

Esperar  500 milisegundos

¿Podrías hacer lo mismo con la lectura de la **luz del sensor LDR** ?