

PROGRAMANDO CON BITBLOQ

VI

COMUNICACIÓN CON OTROS
DISPOSITIVOS: EL PUERTO SERIE

¿Qué es el puerto serie?

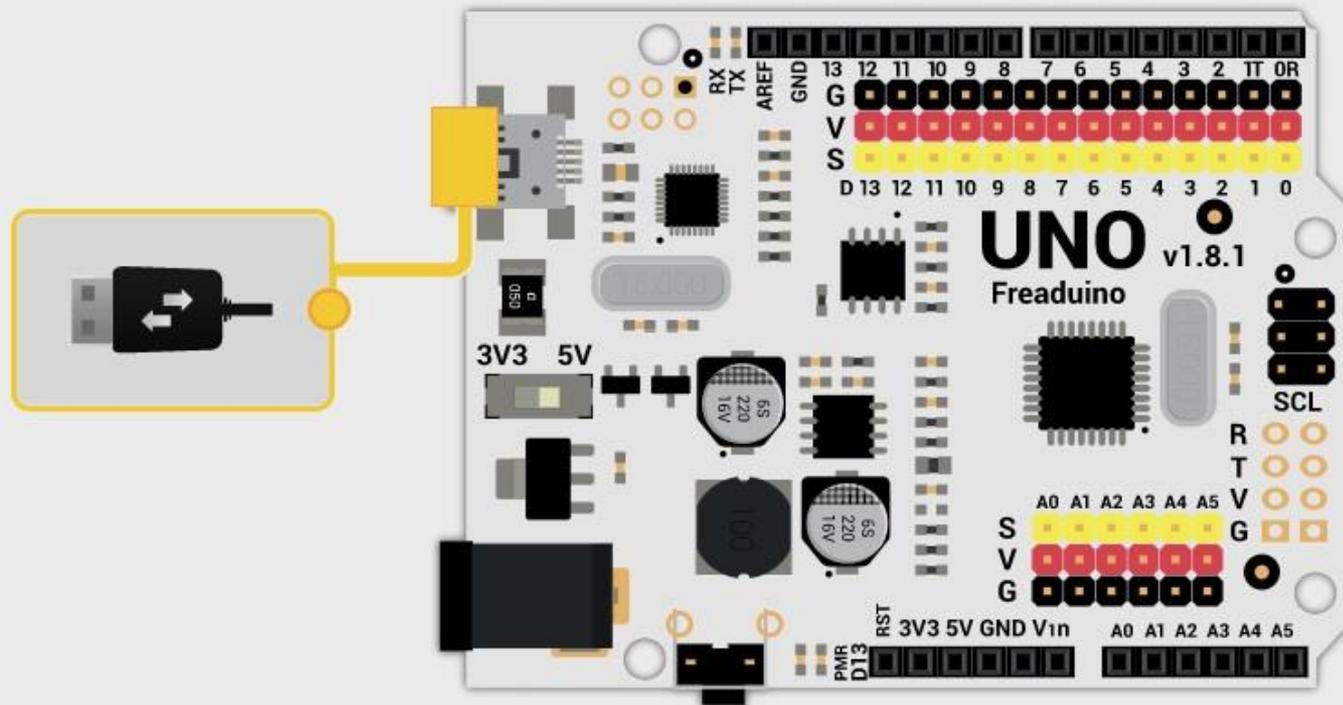
El puerto serie es el medio que tiene tu placa controladora para comunicarse con otro aparato como el ordenador o tu móvil. Es muy útil para comprobar de un vistazo el valor de un sensor o una variable, ver el estado de tu programa y, en definitiva, corregir y entender mejor los posibles errores que éste tenga.

Para usar el puerto serie con el ordenador deberás conectar tu placa mediante un cable USB. En la pestaña *Hardware* selecciona el icono del puerto serie y arrástralo, se conectará automáticamente al puerto USB de tu placa.

Nombre puerto_serie

Baudrate

9600



Vamos a hacer dos retos; el primero es enviar un mensaje sencillo al ordenador

HOLA CARACOLA !!!

Dentro de la pestaña **Componentes** encontrarás el bloque *enviar*. Este bloque te permitirá enviar información mediante el puerto serie a tu ordenador. Clica sobre la pestaña **Texto** y selecciona el bloque *texto*. Escribe el mensaje *HOLA CARACOLA!!!*, añade una espera de un segundo y carga el programa en tu placa.



The image shows a screenshot of a microcontroller IDE's block programming interface. At the top, there is a grey header bar with a minus sign and the text "Bucle principal (Loop)". Below this, a red block labeled "Enviar" is connected to a blue block labeled "Esperar". The "Enviar" block has a dropdown menu set to "puerto_serie", a text input field containing "HOLA CARACOLA", and another dropdown menu set to "Con salto de línea". The "Esperar" block has a text input field containing "1000" and the unit "ms".



Proyecto sin título

Proyecto guardado correctamente

Archivo

Editar

Ver

Compartir

Ayuda



Bloques



Mostrar Serial Monitor



Configuración web2board



Mostrar Plotter



Mostrar Visor



— Bucle principal (Loop)

puerto_serie ▾

Enviar

" HOLA CARACOLA "

Con salto de línea ▾

Esperar

1000

ms



Proyecto sin título

Proyecto guardado con

Archivo

Editar

Ver

Compartir

Ayuda



Bloques

Código

— Bucle principal (Loop)

puerto_serie ▾

Enviar

"

HOLA

SERIAL MONITOR



Send text to board

Enviar

```
HOLA CARACOLA  
HOLA CARACOLA  
HOLA CARACOLA  
HOLA CARACOLA
```

Vamos a por el segundo reto: vamos a leer los datos de un sensor para poder programar una acción concreta a partir de un dato concreto.

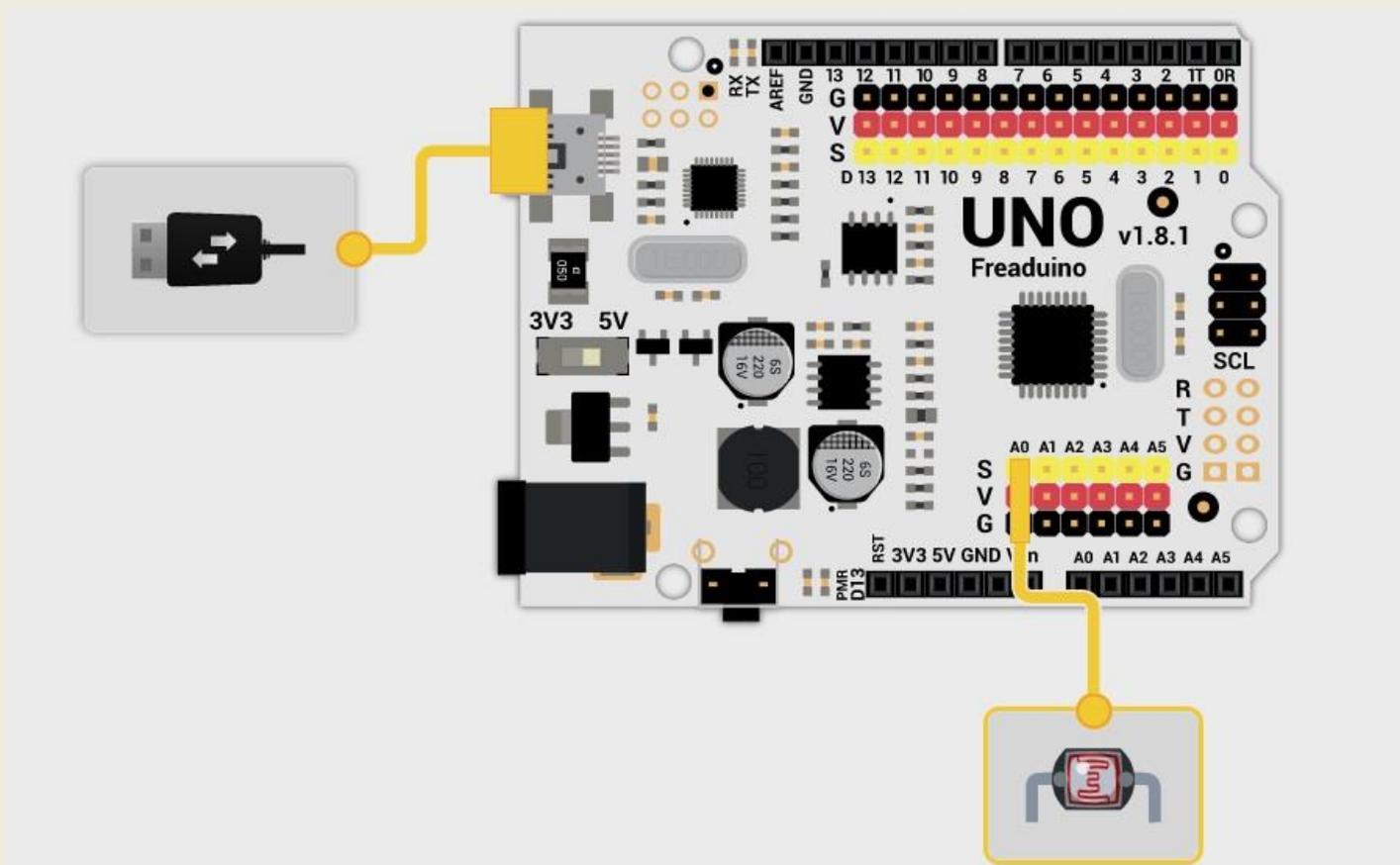
Vamos a trabajar con un sensor de luz:

El sensor de luz es un componente analógico y como tal ofrece diferentes datos, en éste caso la cantidad de luz que recibe en un momento y que puede ser variable (típicamente da entre 0 y 500).

Vamos a hacer que el sensor de luz envíe datos desde la placa al ordenador, y en el ordenador vamos a visualizarlos.

Cómo lo hacemos ???

- Vamos a conectar el sensor de luz en el pin ANALÓGICO 0
- Lo conectamos al hardware de la placa



- Ahora la programación

— Bucle principal (Loop)

puerto_serie_2 ▾

Enviar

Con salto de línea ▾

— Bucle principal (Loop)

puerto_serie_2 ▾

Enviar

Leer

sensor_de_luz ▾

Con salto de línea ▾

— Bucle principal (Loop)

puerto_serie_2 ▾

Enviar

Leer

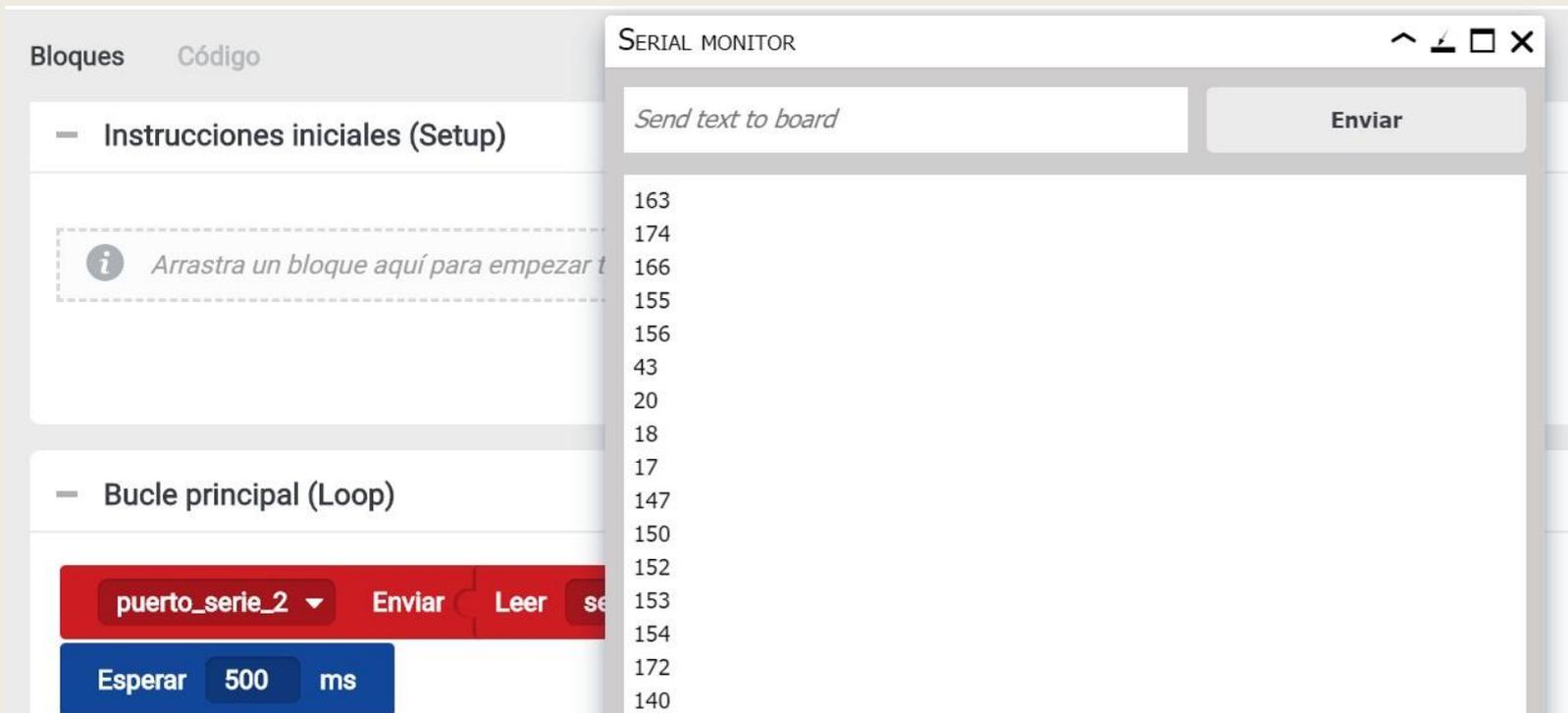
sensor_de_luz ▾

Con salto de línea ▾

Esperar 500 ms

Ahora el programa va a mandar la lectura del puerto serie al ordenador cada medio segundo.

- Compilamos
- Enviamos a la placa
- Como en el reto anterior abrimos : ver / mostrar serial monitor, para empezar a ver los datos recogidos.
- Prueba a tapar y /o sobreiluminar el sensor para comprobar la variación de datos



The image shows a screenshot of an IDE interface. On the left, the code editor is visible with two sections: "Instrucciones iniciales (Setup)" and "Bucle principal (Loop)". The "Bucle principal (Loop)" section contains a red block for sending data to the serial port (puerto_serie_2) and a blue block for waiting 500 ms. On the right, the "SERIAL MONITOR" window is open, displaying a list of numbers: 163, 174, 166, 155, 156, 43, 20, 18, 17, 147, 150, 152, 153, 154, 172, and 140. The window has a text input field with the placeholder "Send text to board" and an "Enviar" button.