

Arduino incorpora una conexión serie que permite conexión con el PC (o con otros muchos dispositivos). Esta es la misma conexión se utiliza para subir el programa al Arduino. Utilizando esta conexión podemos enviar información desde Arduino al PC y al revés.

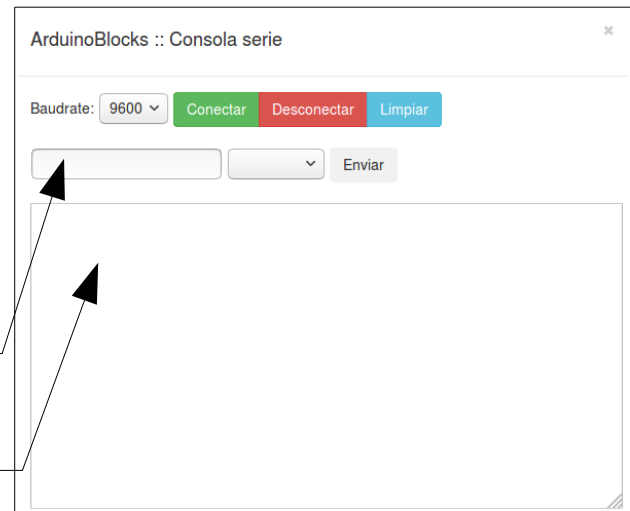
Consola de ArduinoBlocks

Para mostrar la información recibida en el PC y poder enviar datos al Arduino lo más fácil es utilizar un terminal serie o **consola serie**.

La consola serie permite enviar datos desde el PC a la placa Arduino. Y al revés, permite recibir y visualizar los datos recibidos desde la placa Arduino.

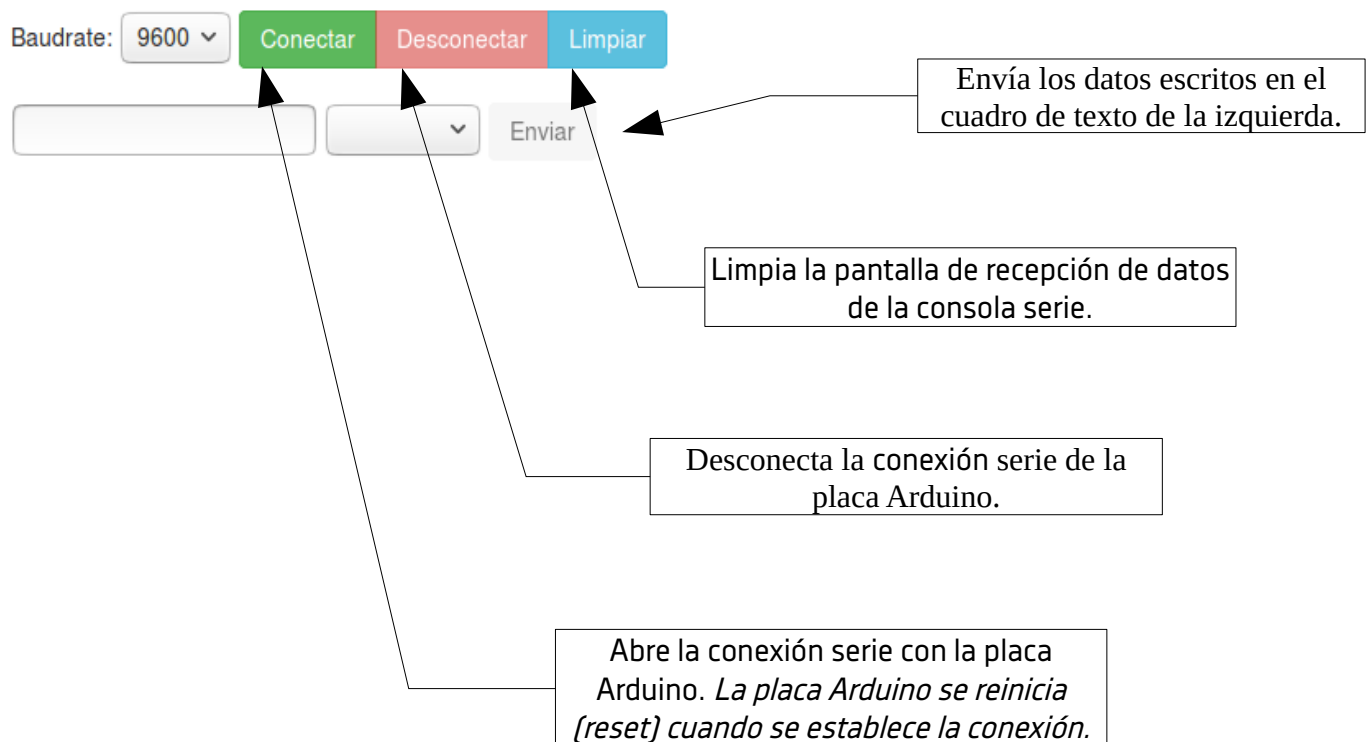
Aquí escribimos para enviar a Arduino

Aquí aparecen los datos recibidos desde Arduino



Opciones de la consola serie:

ArduinoBlocks :: Consola serie



Si no queremos utilizar la consola serie de ArduinoBlocks podemos utilizar cualquier programa de terminal serie para nuestro PC, incluido el propio de Arduino IDE.

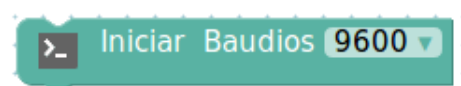
Ejemplo:

<https://sourceforge.net/projects/realterm/>

<https://sourceforge.net/projects/hypeterminal/>

BLOQUES

Bloques de puerto serie en ArduinoBlocks



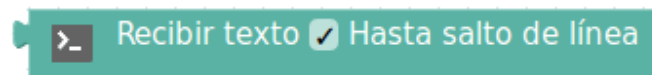
Inicia la configuración de la conexión serie. Debe ser igual en el PC (consola serie) que en el Arduino.



Envía un texto o el valor de una variable a través de la conexión serie. Los datos enviados podemos visualizarlos en la consola de ArduinoBlocks.



Este bloque permite comprobar si existen o no datos pendientes de leer del puerto serie, es decir, si desde la consola del PC se han enviado datos y están pendientes de leer lo sabremos con este bloque.

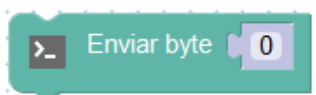


Recibe un texto desde el puerto serie. Normalmente este texto habrá sido enviado desde la consola de ArduinoBlocks



Recibe un texto y lo interpreta como un valor numérico. Normalmente este valor habrá sido enviado desde la consola de ArduinoBlocks.

IMPORTANTE: Si no se recibe nada, se supera el timeout o se recibe un número no válido devolverá "0"

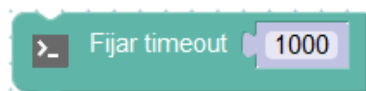


Envía un dato por el puerto serie como un dato numérico de tamaño de un byte (8 bits). Por tanto el valor será entre 0 y 255



Recibe un valor por el puerto serie de tamaño de un byte (8 bits).

El valor recibido será entre 0 y 255



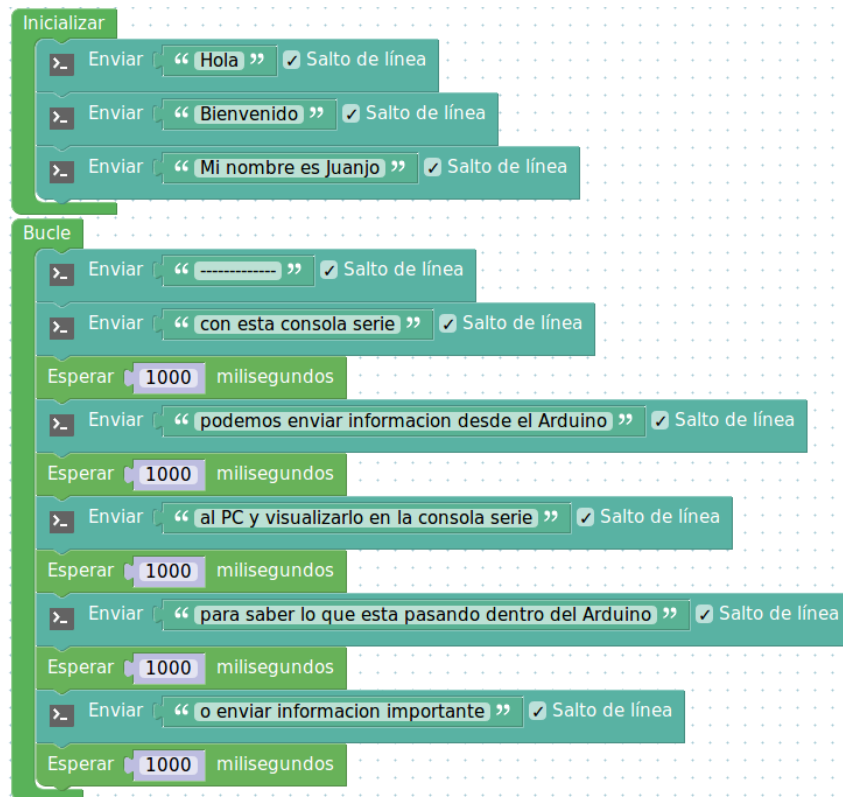
Establece el valor máximo de tiempo de espera a la hora de recibir y enviar un dato por el puerto serie.

Por defecto es de 1000 ms.

PRÁCTICA 6.1 Enviar mensajes desde Arduino

CÓDIGO DE PROYECTO:

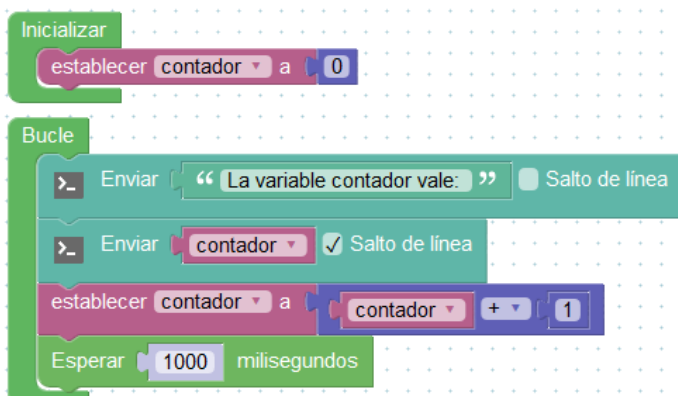
El programa enviará mensajes de texto desde Arduino, para visualizar los datos recibidos desde la conexión serie utilizaremos la consola que incorpora ArduinoBlocks.



PRÁCTICA 6.2 Enviar el valor de una variable

CÓDIGO DE PROYECTO:

Vamos a ver como enviar el valor de una variable, en este caso una variable que va incrementándose y mostramos en la consola serie su valor.

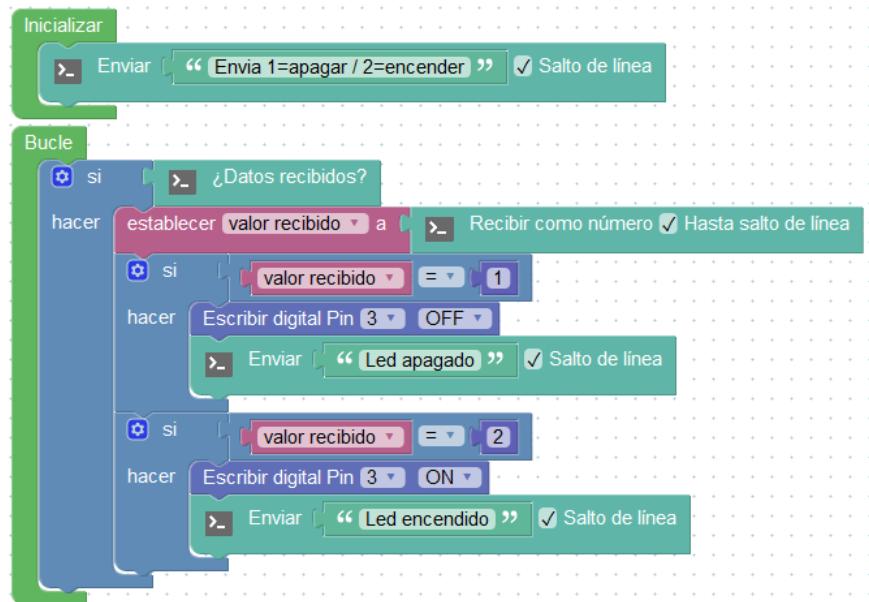
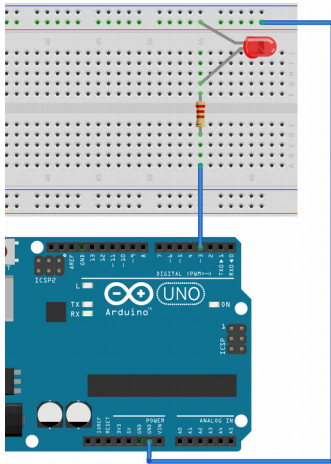


La variable contador vale: 1.00
La variable contador vale: 2.00
La variable contador vale: 3.00
La variable contador vale: 4.00
La variable contador vale: 5.00
La variable contador vale: 6.00
La variable contador vale: 7.00
La variable contador vale: 8.00
La variable contador vale: 9.00

PRÁCTICA 6.3 Encendido de un led desde el PC

CÓDIGO DE PROYECTO:

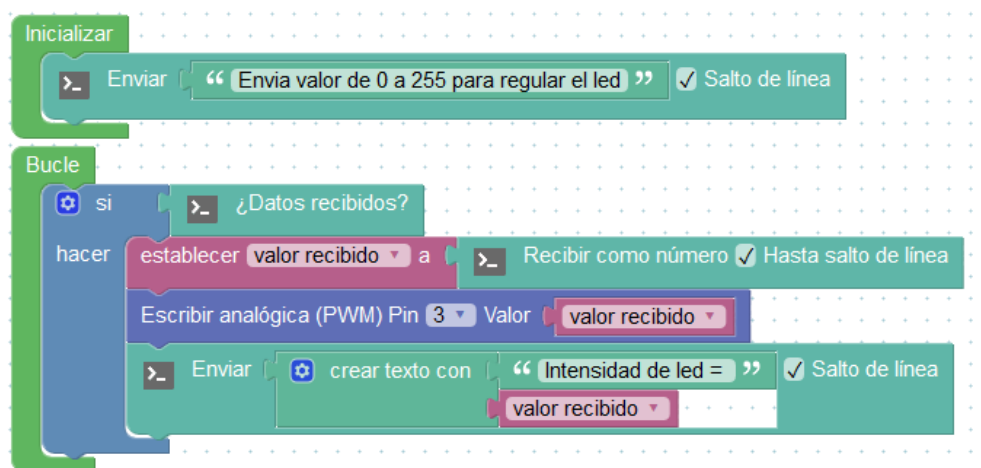
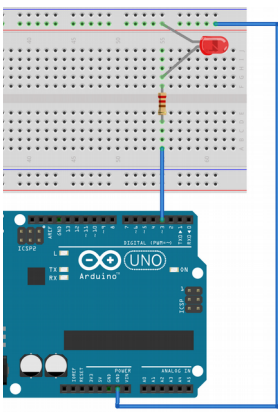
Al recibir el valor 1 desde la consola apagaremos el led, al recibir el 2 lo encenderemos.



PRÁCTICA 6.4 Regulación de intensidad de led desde PC

CÓDIGO DE PROYECTO:

Conectar un led al pin 3. Recibir un número desde el ordenador a través del puerto serie (como en el programa 4). El número recibido debe estar entre 1 y 255. El número recibido se escribe en la salida analógica (PWM) del pin 3. De esta forma podemos regular la intensidad del led desde la consola del ordenador.



PRÁCTICA 6.5 Juego: Adivina el número

CÓDIGO DE PROYECTO:

Realizaremos un juego donde la placa Arduino “pensará” un número al azar entre 1 y 100. Desde la consola iremos diciendo números y nos dirá si el número secreto es mayor o menor hasta que lo adivinemos y nos muestre el número total de intentos que hemos usado.

