

### 1.2.16. Sensor Analógico: Potenciómetro

En esta actividad vamos a utilizar un potenciómetro como ejemplo de sensor analógico. Los potenciómetros son utilizados en multitud de aplicaciones, como por ejemplo para cambiar el volumen de la radio. En robótica también son muy utilizados. Por ejemplo los servos que vimos en la actividad X llevan dentro potenciómetros para saber en qué posición está el eje del motor, es decir, actúan como sensores de posición.

Vamos a aprender a utilizar un potenciómetro a la vez que aprendemos a utilizar señales analógicas en Arduino. Una señal analógica es una señal que toma cualquier valor entre un rango determinado. Por ejemplo en Arduino las señales analógicas pueden tomar cualquier valor en 0v y 5v (por ejemplo 3.56) mientras que las digitales pueden valer 0v (LOW) o 5v (High).

Si hacemos pasar una corriente por el potenciómetro el valor de tensión variará en función de si lo giramos hacia la izquierda o hacia la derecha. Esto es debido a que, como sabes, un potenciómetro es una resistencia variable. Por lo tanto, si conectamos un potenciómetro, como el del kit de BQ, a un pin de Arduino la salida del potenciómetro (o lo que es lo mismo, a la entrada del pin) estará entre 0v y 5v.

En realidad vamos a realizar dos actividades, una primera en la que vamos a intentar leer el valor de un potenciómetro en un programita en Bitbloq y mostrarlo en nuestro ordenador a través de un monitor serie, y una un poco más avanzada en la que utilizaremos un potenciómetro para controlar un servo.

#### 1.2.16.1. Lectura de un potenciómetro

##### Componentes:

- Placa ZUM BT o otra Arduino UNO compatible
- Potenciómetro del kit de robótica de BQ u otro cualquiera.
- Cable USB para la comunicación serie con nuestro PC.

##### Conexionado

Conectaremos el potenciómetro a cualquier pin de entrada analógica de Arduino. Hay 6 entrada analógicas en las placas compatibles con Arduino UNO (como la ZUM BT). Están numeradas de la A0 a la A6 (recuerda que los pines que están numerados del 0 al 13 son entradas y salidas digitales, algunas de ellas con pwm lo que emula una salida analógica). Por ejemplo realizaremos la siguiente conexión:

- Potenciómetro --, Entrada analógica A3
- Cable USB del PC al Arduino UNO compatible (por ejemplo ZUMBT)

Una vez realizada la conexión física procederemos a crear un nuevo proyecto en Bitbloq y pondremos en él los tres componentes (placa de control, potenciómetro y cable USB) como en la Figura 1.2.16-1

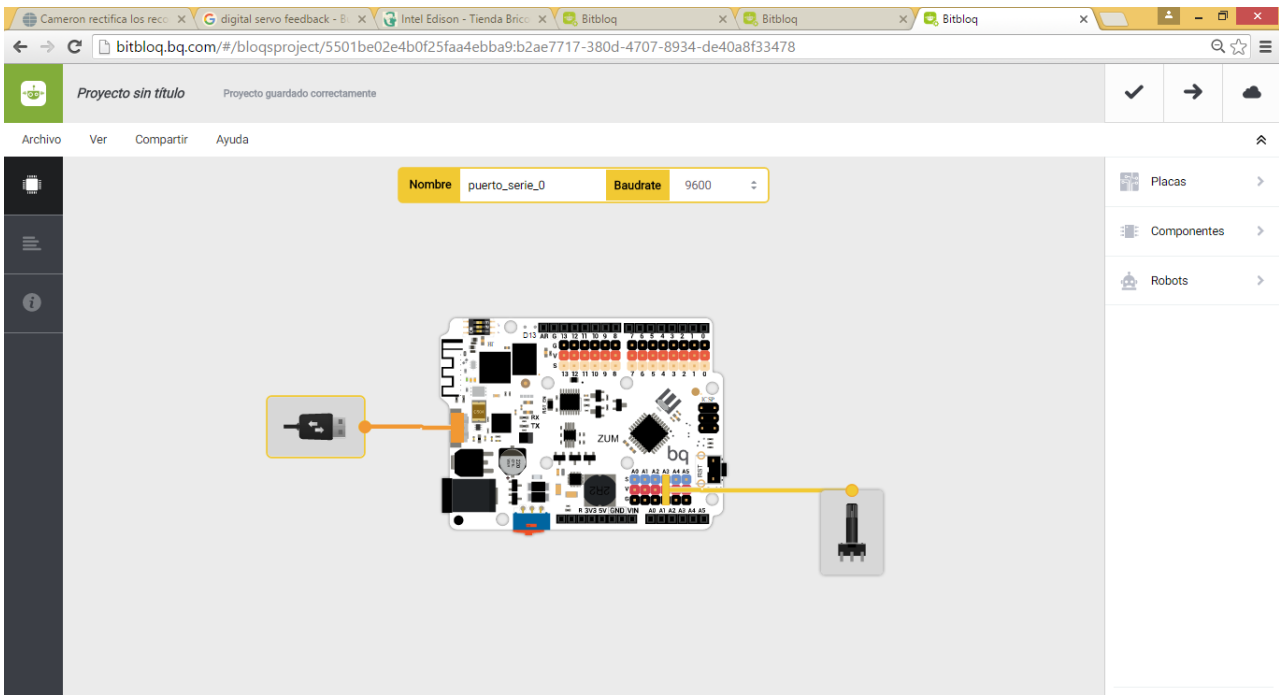


Figura 1.2.16-1 Conexión de potenciómetro en Bitbloq

## Programación

El componente potenciómetro en Bitbloq nos ofrece un bloque leer(potenciómetro\_0). Podemos utilizar este bloque para enviar lo leído por el puerto serie utilizando el bloque enviar(Puerto\_serie\_0) como aparece en la Figura 1.2.16-2

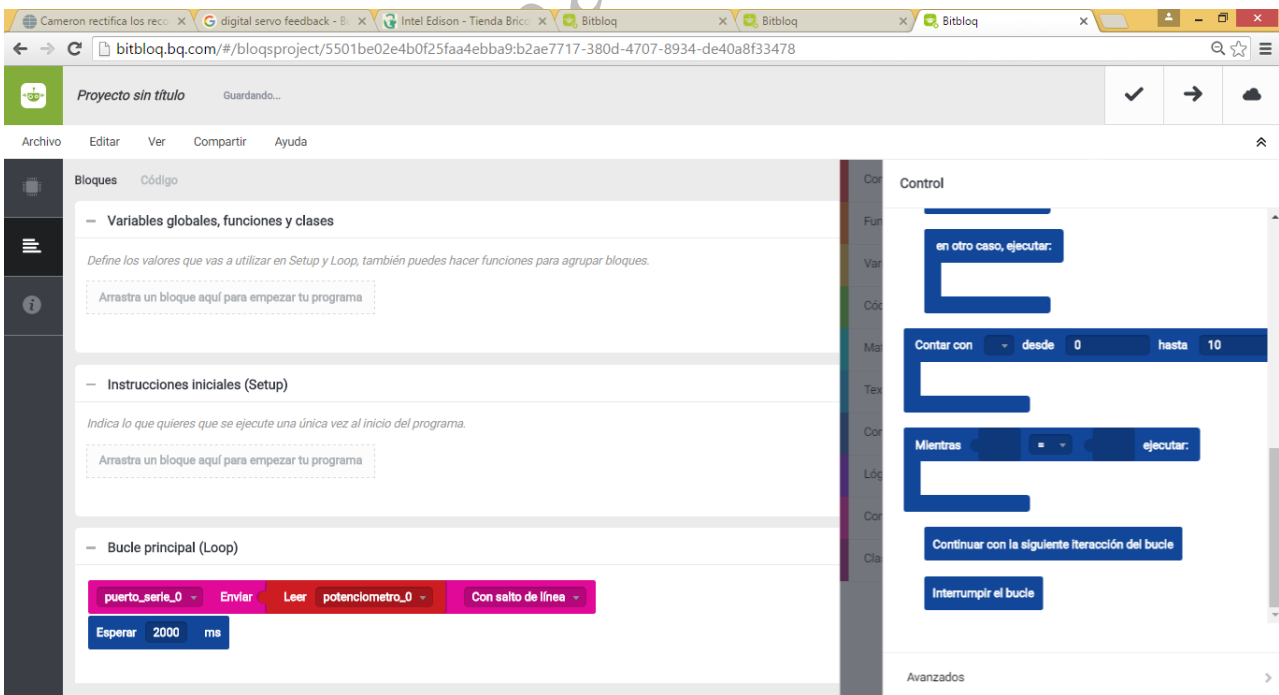


Figura 1.2.16-2 Código de bloques de la actividad de lectura del potenciómetro

## Libro de Actividades de Robótica Educativa

De esta manera estaremos enviando por el puerto serie cada 2 segundos (2000ms) el valor leído del potenciómetro. Si cargamos el programa en nuestro Arduino UNO compatible y conectamos con un cable USB la placa a nuestro PC podremos utilizar el monitor serie para ver dichos valores (ver figura)

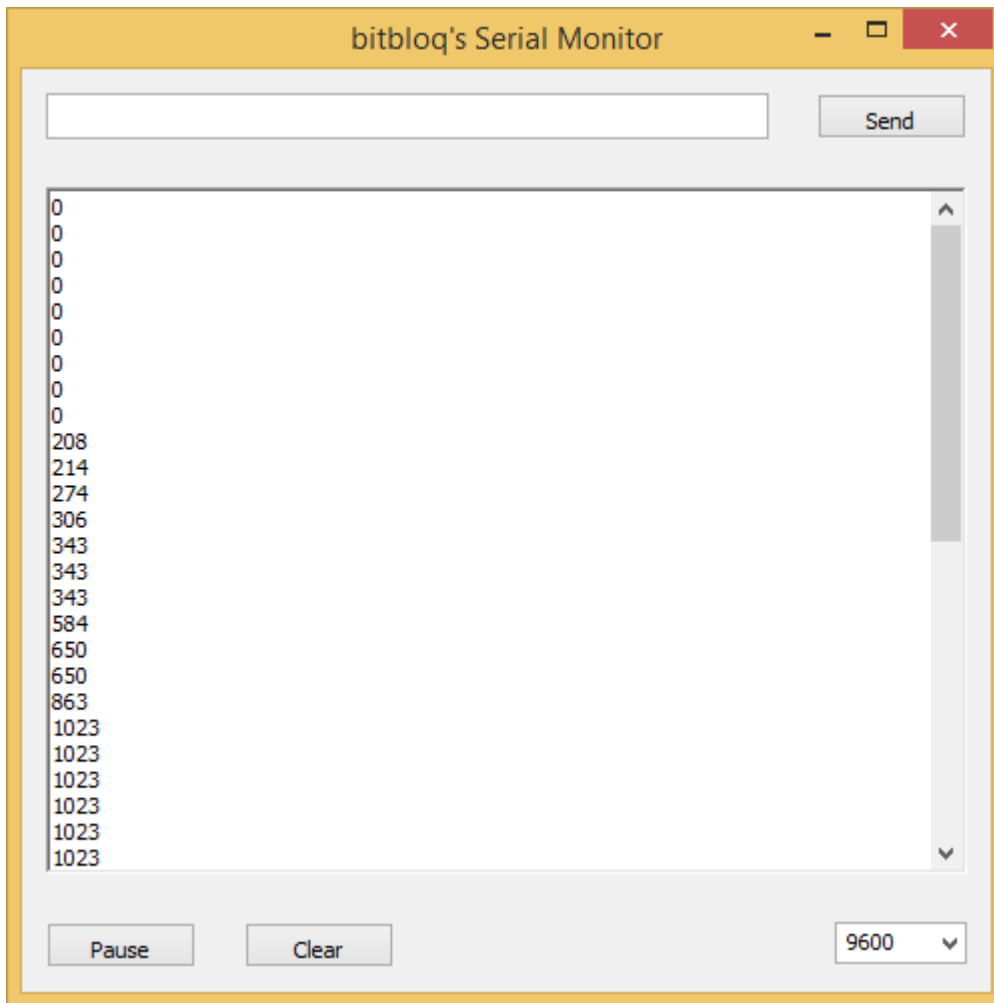


Figura 1.2.16-3 Salida por el monitor serie de los datos del potenciómetro

Como vemos los valores leídos van del 0 al 1023. Como sabes estos valores son datos que han sido convertidos de analógico a digital ya que los procesadores como el que tiene Arduino, o incluso los ordenadores, no funcionan con datos analógicos, sino que funcionan con datos digitales. En realidad 0 (0000000000 en binario o lenguaje máquina, que es lo que entiende un procesador) corresponderá a 0v y 1023 (1111111110 en binario) a 5v. Esto es así porque nuestra placa tiene un conversor analógico a digital de 10bits ( $2^{10}=1024$ ).

### 1.2.16.2. Control de un servo con un potenciómetro.

En esta actividad vamos a hacer girar un servo cuando giramos un potenciómetro. Será como tener un mando en el que un ángulo girado en el potenciómetro implica el mismo giro en el servo.

## Componentes

## Libro de Actividades de Robótica Educativa

- Placa de control ZUMBT o Arduino UNO compatible
- Servo de posición
- Potenciómetro

## Conexionado

Realizamos el siguiente conexionado sobre nuestra placa de control:

- Servo de posición → Pin 7 Arduino
- Potenciómetro → Pin A1 Arduino
- Cable USB PC → Arduino

Procedemos a abrir un nuevo proyecto en Bitbloq y añadimos los componentes tal como se muestra en la Figura 1.2.16-4.

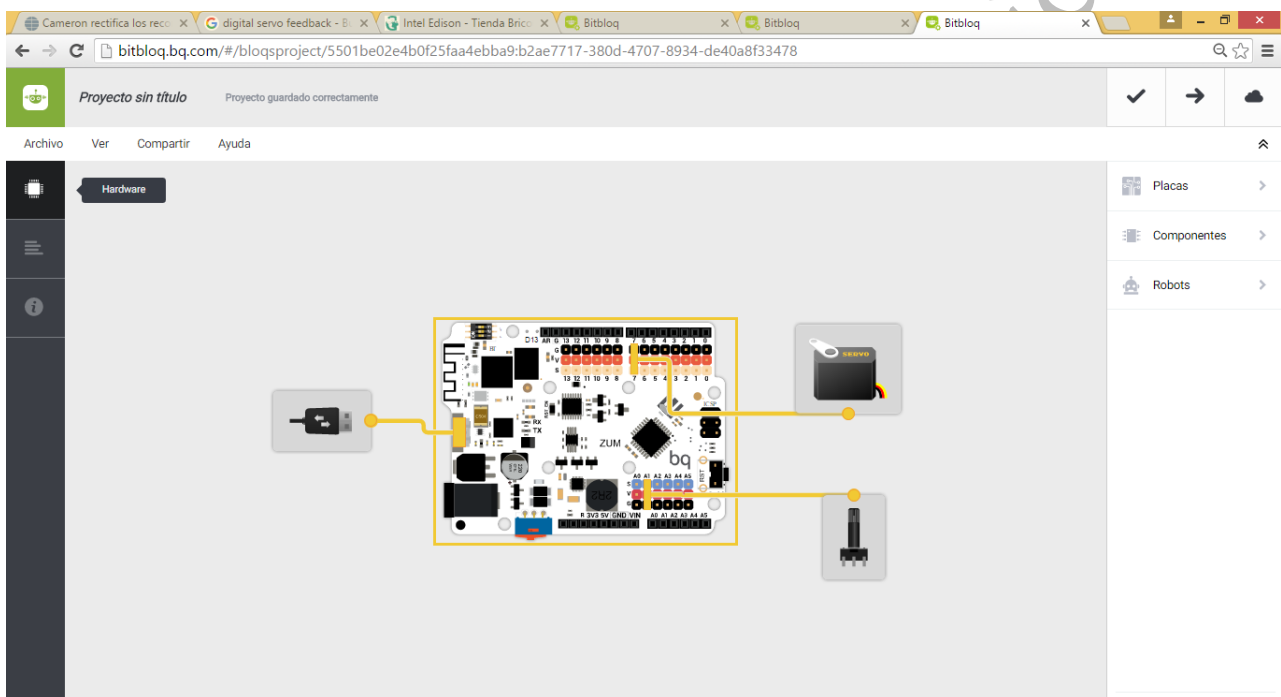


Figura 1.2.16-4 Conexionado en bitbloq para el control de un servo con un potenciómetro

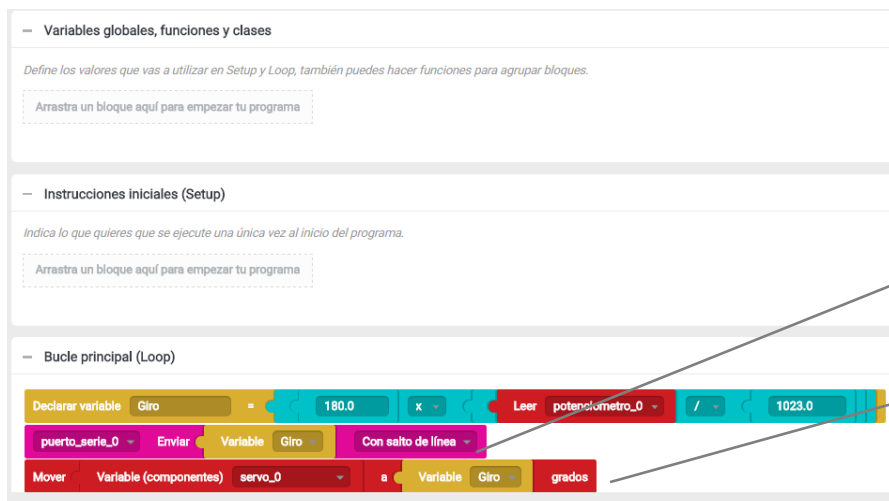
## Programación

Vamos a añadir los bloques que permitan mover el servo en función de lo que se mueve el potenciómetro. Como un servo se mueve de 0° a 180° y la lectura de un potenciómetro puede tomar valores de 0 a 1023 deberemos realizar una conversión entre dichos rangos. Por eso, el primer bloque que añadiremos es la declaración de una variable con la siguiente ecuación:

$$Giro_{servo} = 180 * \frac{lectura_{potenciometro}}{1023}$$

Los siguientes bloques serán para ordenar al servo que realiza ese giro y para mostrar por el puerto serie dicho valor. El código de bloques resultante se muestra en la Figura 1.2.16-5

## Libro de Actividades de Robótica Educativa



Cálculo de la variable de giro en función de la medición del servo

Envío por el puerto serie de lo que va a moverse el servo

Comando para mover el servo

Figura 1.2.16-5 Código de bloques para control de un servo con un potenciómetro

El código en Arduino resultante se muestra en la siguiente figura.

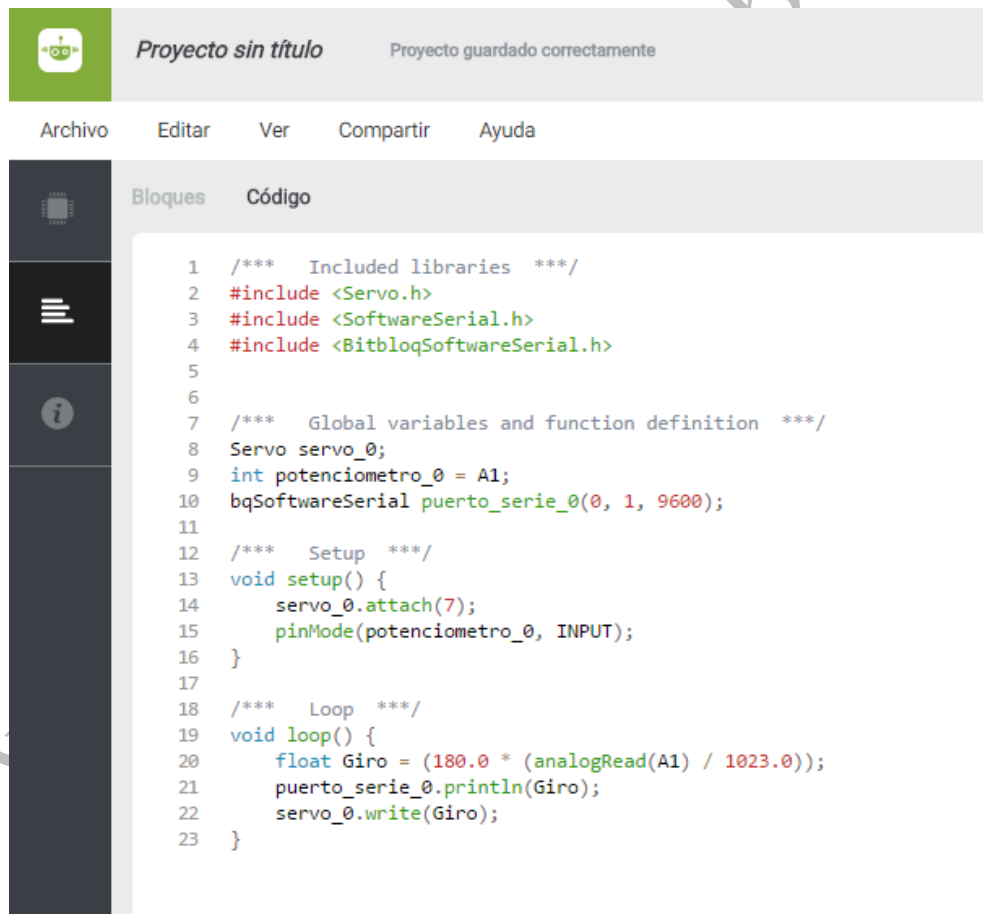


Figura 1.2.16-6 Código en Arduino de la actividad

Como puede observarse, en la línea 14 asignamos el servo al pin 7 y en la 15 decimos que el pin A1 será una entrada (INPUT).

En la 20 realizamos el cálculo de la formula para saber cuanto tiene que girar el servo, utilizando como dato de entrada la lectura analógica (analogRead) del pin A1. En la 21 enviamos por el puerto serie dicho valor y en la 22 le decimos al servo que realice el giro (con servo.write).