

1.2.7. Robot Loro

En esta actividad se pretende afianzar los conocimientos sobre la comunicación serie y el uso de variables. Vamos a realizar un programa que lee lo que se envía por puerto serie a la placa desde el PC, y lo devuelve por el puerto serie.

Mediante la consola del monitor serie es posible enviar datos a la placa desde el PC, se empleará esa herramienta para mandar el texto que se desea reproducir.

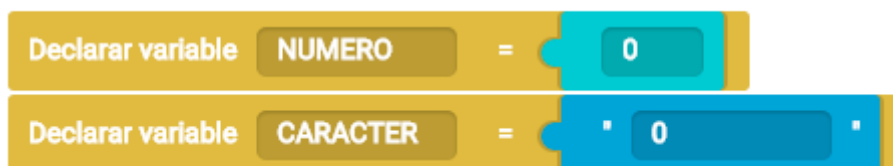
Componentes necesarios para la actividad:

- Placa ZUM BT, Arduino o compatible.
- Cable USB para conectar la placa al PC

El conexionado hardware es exactamente igual al del ejercicio anterior, tan solo es necesario el bloque de comunicación serie.

Primero vamos a declarar las variables que vamos a usar, pero antes es importante explicar la importancia del tipo de los datos. En la Figura 1.2.7-1 se muestra como se declaran dos variables, una llamada “NUMERO” a la que se asigna el valor 0 y que por tanto será de tipo decimal y otra llamada “CARACTER” a la que se le asigna el caracter 0 por tanto será una variable de tipo cadena de caracteres. Esto es importante ya que los datos que se envían por puerto serie son siempre caracteres. Si, como en la parte inferior de la imagen, tratamos de asignar la lectura del puerto serie a la variable “NUMERO” BitBloq nos lo impedirá, mientras que cuando se lo asignamos a la variable “CARACTER” si que funciona ya que los dos datos son del mismo tipo.

– Variables globales, funciones y clases



Definición de variable de tipo numérico

Definición de variable de tipo cadena de caracteres

– Bucle principal (Loop)



Figura 1.2.7-1. Tipos de datos de variables en BitBloq.

Volviendo a la actividad. Vamos a comenzar con la declaración de variables. En la Figura 1.2.7-2 se muestra la declaración de variables, como podemos observar se ha creado una variable llamada “Serie” y se le ha asignado un carácter vacío para inicializarla

– Variables globales, funciones y clases



Figura 1.2.7-2. Declaración de variables en BitBloq de la actividad Robot Loro.

A continuación vamos a introducir una novedad, se va a utilizar código en “Instrucciones Iniciales”. Aquí vamos a poner líneas de código que solo se ejecutan una vez y no de forma cíclica como en el “Bucle principal”. Al empezar el programa vamos a enviar por puerto serie la frase “Hola. Soy tu robot loro”. La línea de código debe quedar como la que se muestra en la Figura 1.2.7-3.

– Instrucciones iniciales (Setup)



Figura 1.2.7-3. Uso de instrucciones iniciales en BitBloq.

Ahora pasamos a la programación del “Bucle principal”. Para saber cuando debemos enviar por puerto serie una frase, tenemos que saber si la hemos recibido. El programa se inicia con la asignación de lo que se lee por puerto serie a la variable “Serie”.

Tras ello encontramos una sentencia condicional que comprueba si la longitud de la cadena de caracteres que contiene “Serie” es mayor que 0, o lo que es lo mismo si, si hay al menos un carácter es que he recibido algo. Para ello usamos el bloque “Longitud” que se encuentra en la pestaña “Texto”.

Si la condición se cumple, la cadena de caracteres contenido en “Serie” se envía por el puerto. Podemos ver el programa completo en la Figura 1.2.7-4.

– Bucle principal (Loop)



Figura 1.2.7-4. Código en BitBloq de la actividad Robot Loro

Libro de Actividades de Robótica Educativa

Ahora solo es necesario cargar el programa y abrir la consola de comunicación serie. En la parte superior existe una línea que nos permite enviar mensaje por el puerto. Los mensajes que enviamos son recibidos por la placa y devueltos de nuevo por puerto serie.

El código en Arduino es el siguiente:

```
/** Included libraries */
#include <SoftwareSerial.h>
#include <BitbloqSoftwareSerial.h>

/** Global variables and function definition */
BqSoftwareSerial puerto_serie_0(0, 1, 9600);
String Serie = " ";

/** Setup */
void setup() {
  puerto_serie_0.println("Hola. Soy tu robot loro");
}

/** Loop */
void loop() {
  Serie = puerto_serie_0.readString();
  if (Serie.length() > 0) {
    puerto_serie_0.println(Serie);
  }
}
```

www.automatica.com