

Iniciación a la robótica con LEGO® MINDSTORMS, 14ª entrega

Múltiple calibración de sensores de luz

Por Koldo Olaskoaga

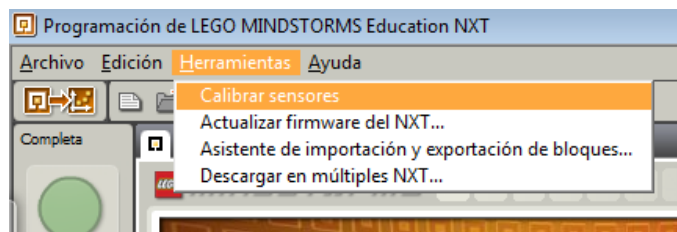
Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas [1]. Estas magnitudes pueden ser intensidad lumínica, distancia, aceleración, inclinación, temperatura, presión sonora...

En el mundo real, las condiciones del entorno pueden afectar a las lecturas de los sensores y en ocasiones pueden dar verdaderos dolores de cabeza. En el caso de los sensores de luz, las condiciones de iluminación pueden hacer que un robot que realizaba perfectamente su tarea en el espacio en el que se diseñó, sea incapaz de realizar su tarea en otro lugar, lo que en un lugar el sensor veía como blanco en otro lo verá como gris.

Así que, en la situación de trabajo real, es necesario decirle al robot qué es blanco y qué es negro, es decir, calibrar el sensor.

Calibración básica de un sensor de luz

Hay dos métodos estándar para calibrar un sensor de luz en NXT-G. El primero es por medio de la opción **Calibrar sensores** que podemos encontrar en el menú **Herramientas** de NXT-G y la segunda creando un pequeño programa con el bloque de programación **Calibrar** del menú **Avanzado**.



Estos dos métodos permiten una única calibración para todos los sensores de luz conectados al NXT, así que si deseamos una distinta calibración para diferentes sensores o dos diferentes para uno de ellos tendremos que recurrir a otra estrategia; lo haremos utilizando la capacidad que tiene el NXT de registrar datos en un archivo y después leerlos.

Antes de entrar en cómo hacerlo, vamos a ver cómo utilizar archivos.

Uso de archivos con NXT-G

NXT-G permite almacenar información en archivos de texto. Estos archivos se pueden leer desde el mismo u otro programa y ser transferidos al ordenador para su posterior tratamiento.

Esta información puede ser tanto de tipo numérico como textual, por ejemplo, las lecturas de un sensor...

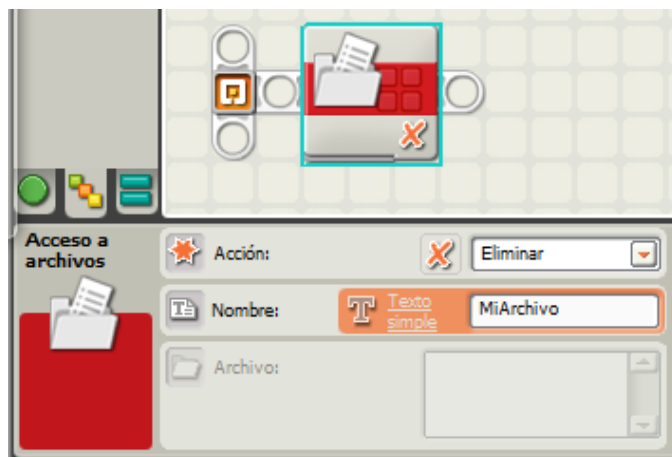
Con carácter general, los pasos para utilizar archivos son los siguientes:

- 1- Escribir en el archivo
- 2- Cerrar el archivo
- 3- Leer los datos.

En todos los casos se utiliza el mismo bloque de programación, **Acceso a archivos**, configurado del modo que se necesite en cada caso.

Crear un archivo

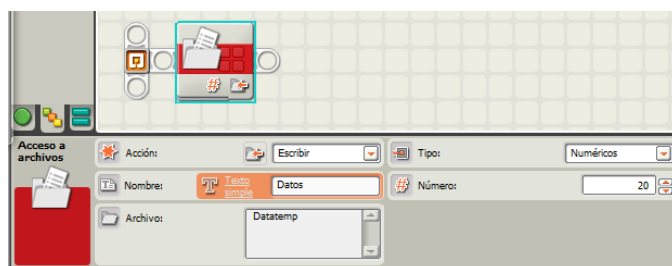
El primer paso será crear el archivo. Dado que probablemente ejecutaremos el programa en más de una ocasión, lo primero que habrá que hacer será eliminar el archivo creado previamente, en caso contrario, añadirá los nuevos datos al archivo antiguo.



En la imagen se puede ver el bloque **Acceso a archivos** configurado para que elimine el archivo de nombre **MiArchivo**.

Una vez eliminado el archivo podemos crear el nuevo y comenzar a escribir en él. No hay un modo de configurar el bloque específico para la creación de un archivo, el archivo se crea cuando se escribe el primer dato en él.

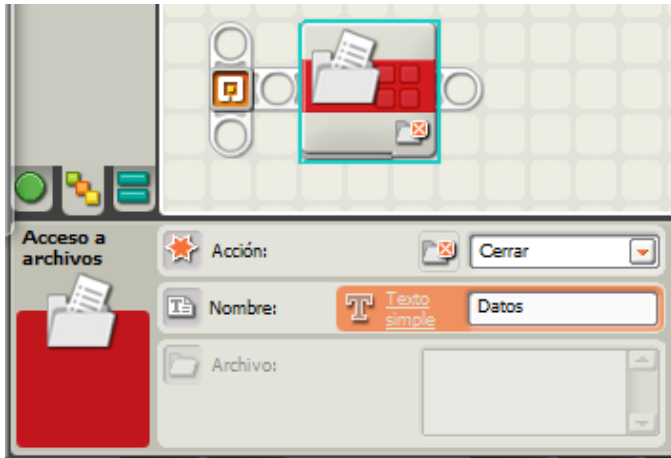
Los datos que puede almacenar un archivo pueden ser de tipo texto o numérico. El bloque de la imagen siguiente escribe un valor numérico en un archivo denominado **Datos**.



Utilizaremos este dato tantas veces como sea necesario para almacenar nuevos valores en el archivo.

Cerrar el archivo

Antes de leer los datos hay que cerrar el archivo, configurando el bloque Acceso a archivos tal y como se ve en la siguiente imagen.

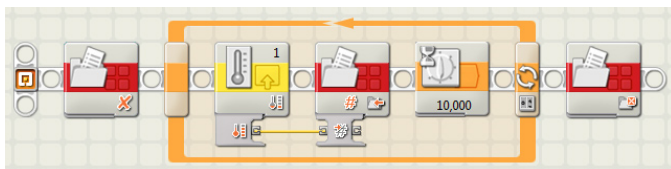


Ejemplo

Veamos cómo se puede aplicar lo anterior por medio de un ejemplo. La idea es registrar la variación de la temperatura a lo largo del tiempo: durante un proceso de enfriamiento o calentamiento, la temperatura atmosférica... Para ello se utiliza el sensor de temperatura para LEGO® MINDSTORMS NXT. Una vez registrados los valores, transferiremos el archivo al ordenador y los convertiremos en un gráfico que ayude en su interpretación.

Esto es algo que se puede hacer de un modo más sencillo con la aplicación de registro de datos de NXT-G Educación, no así con su versión comercial. Por otra parte el uso de archivos permite un registro de datos más flexible.

En el ejemplo, cuyo código puede verse en la siguiente imagen, se mide la temperatura cada segundo durante 100 segundos. Esto significa que se registrarán 100 valores.



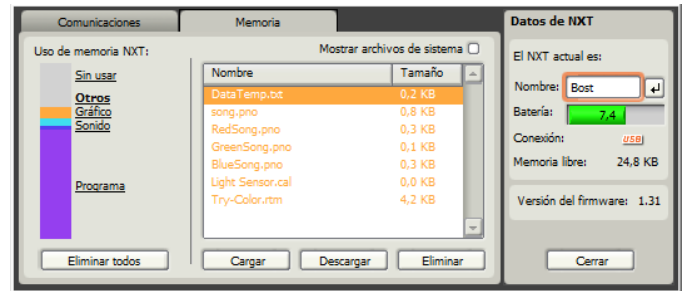
Este programa, tras eliminar el archivo (denominado en este caso DatosTemp), abre un bucle (bloque Bucle configurado en modo Contador) que se repetirá 100 veces. En el bucle lee el sensor de temperatura conectado al puerto 1, registra el valor en el archivo y espera 1 segundo antes de volver al inicio del bucle. Tras la repetición número 100 cierra el archivo y finaliza el programa.

Para transferir el archivo al ordenador primero deberemos abrir la ventana NXT, pulsando sobre el botón superior izquierdo del controlador.

En la ventana NXT seleccionaremos la pestaña **Memoria** y en ella **Otros** y podremos

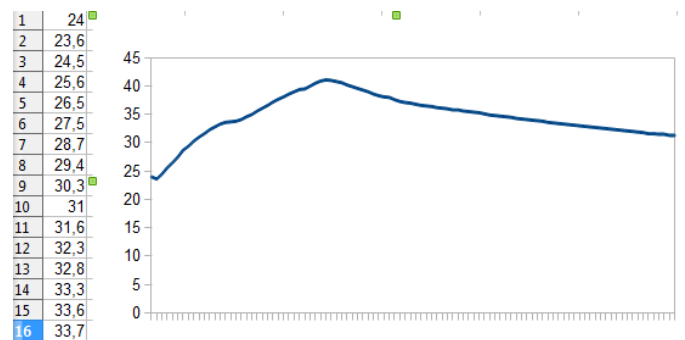


ver varios de los archivos que contiene el NXT, entre ellos el creado en este ejemplo.



Si pulsamos sobre el botón Cargar, podremos transferir el archivo (DataTemp.txt) al directorio que deseemos en nuestro ordenador.

Una vez hecho esto, abriremos el archivo desde una hoja de cálculo, en mi caso desde LibreOffice Calc. Una vez abierto podremos observar que el separador que utiliza NXT-G para los decimales es un punto, por lo que la versión española de LibreOffice Calc no interpretará como valores numéricos las temperaturas; así que tendremos que sustituir los puntos por comas (Buscar y reemplazar). Utilizando la herramienta Insertar Gráfico podremos obtener una representación gráfica de las temperaturas, tal y como puede verse en la siguiente imagen.

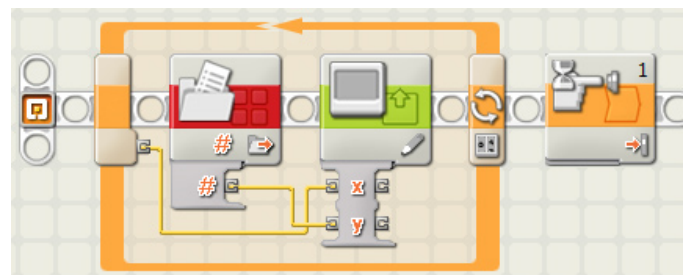


Leer los datos

Veamos ahora cómo podemos leer los datos almacenados previamente en un archivo. Antes de ello cabe señalar que la lectura será secuencial, es decir, se leerán en el orden en que han sido registrados, no podremos leer el tercer dato sin antes haber leído el primero y el segundo.

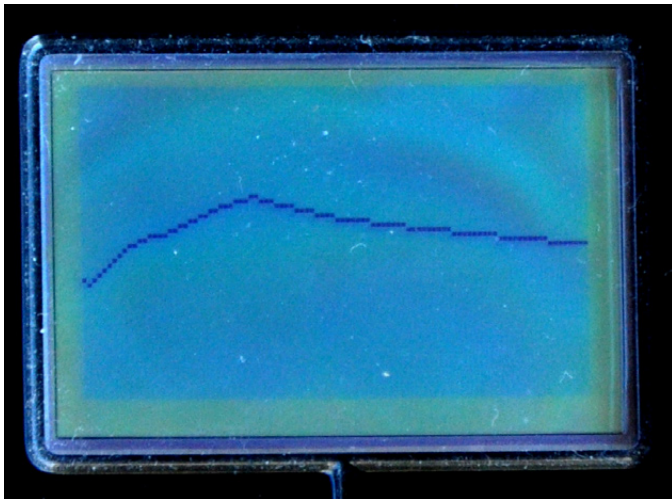
Se pueden leer los datos almacenados en un archivo desde el mismo programa que los registra o desde cualquier otro programa. En este caso lo vamos a hacer desde el segundo programa; vamos a leer los valores almacenados en el archivo DataTemp y los vamos a representar en la pantalla del NXT.

El programa es el siguiente:



El programa consta de un bucle que se repite tantas veces como datos ha de leer. Para permitir que se visualice el gráfico todo el tiempo que se desee se ha añadido un bloque **Espera** configurado para un sensor de contacto. En realidad no hace falta montar ninguno, ya que de este modo se mantendrá la ejecución del programa hasta que se pulse el botón gris del NXT.

El bloque **Visualizar** permite dibujar puntos en la posición que se desee a partir de las coordenadas x e y del punto. En este caso la "x" será el número de orden de la toma de temperatura y la "y" la temperatura. Se configura el bloque **Acceso archivos** en modo lectura y se conecta su toma de salida Número de salida con la toma de entrada "y" del bloque **Visualizar**. La "x" la obtendremos del contador del bucle. De este modo tendremos el mismo gráfico que anteriormente hemos representado en el ordenador, en la misma pantalla del NXT.



Calibración de más de un sensor

Ahora que ya tenemos una idea de cómo utilizar archivos desde NXT-G vamos a ver otro modo de calibrar un sensor.

En los modos de calibración mencionados al principio del artículo, el programa crea automáticamente un archivo y lo lee cada vez que se utiliza un sensor de luz sin que siquiera seamos conscientes de ello. Ahora lo que vamos a hacer es crear un archivo con los datos necesarios para la calibración y utilizarlo cada vez que lo necesitemos.

Crear el archivo de calibración

Para crear el archivo de calibración hemos de leer y registrar los valores máximos y mínimos que pueda leer el sensor en la situación real. Habitualmente se hace posicionando el sensor sobre la zona más clara y a continuación sobre la más oscura, pero aquí lo vamos a hacer de otra manera: moveremos el robot del mismo modo que se moverá cuando realice su tarea y registraremos los valores máximo y mínimo. A continuación los guardaremos en el archivo correspondiente.

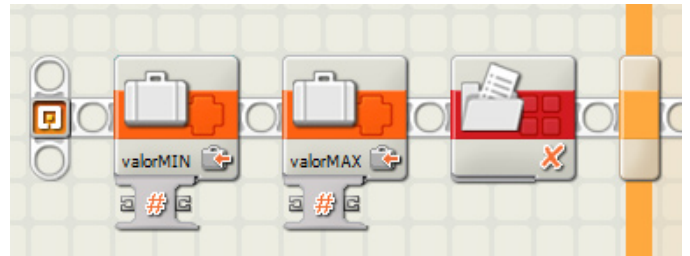
Veamos primero cuáles son los pasos a dar:

- 1- Crear dos variables (**valorMAX** y **valorMIN**) que almacenarán los valores máximo y mínimo de luminosidad leída.
- 2- Inicializar las variables con los valores **valorMAX=0** y **valorMIN=1023** (en lugar de utilizar el valor de lectura porcentual, utilizaremos el valor directo que está

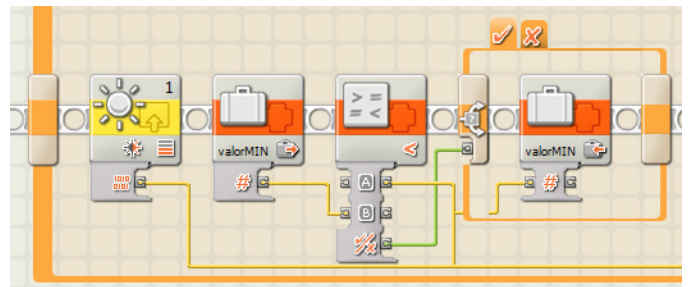
comprendido entre 0 y 1023).

- 3- Eliminar el archivo de calibración (**Calibra1**).
- 4- Iniciar un bucle, que se ejecutará durante 5 segundos, con los pasos siguientes:
 - a. leer el valor del sensor de luz
 - b. comparar la lectura con la variable **valorMAX**, si es mayor que ella, almacenar la nueva lectura en la variable
 - c. comparar la lectura con la variable **valorMIN**. Si es menor que ella, almacenar la nueva lectura en la variable.
- 5- Una vez cerrado el bucle escribir el contenido de la variable **valorMIN** en el archivo de nombre **Calibra1** y a continuación lo mismo con **valorMAX**.
- 6- Cerrar el archivo **Calibra1**.

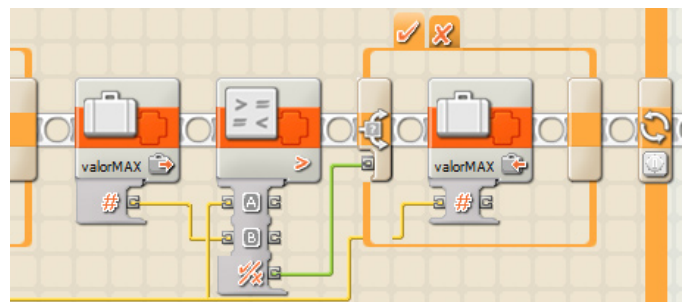
Una vez hecho esto el archivo de calibración estará almacenado en el NXT. Este sería el algoritmo básico, al que si se quiere se puede añadir movimiento, para que en lugar de ser el usuario el que mueve el robot sobre las zonas clara y oscura, sea el mismo robot el que se mueva (así se asegura que el movimiento del robot sobre la línea será completamente real).



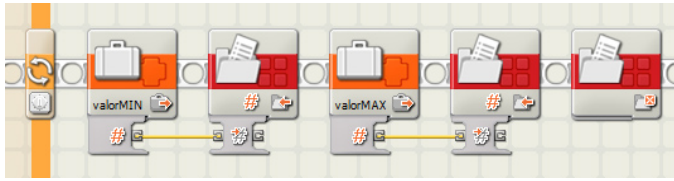
Tras crear las variables vía la opción Definir variables del menú Edición, haremos los pasos 2 y 3 tal y como se ve en la figura anterior.



Se puede ver en este fragmento de código (pasos 3a y 3b) cómo se obtiene el valor directo de la lectura del sensor de luz conectado al puerto 1 y se compara con el valor almacenado en la variable **valorMIN**. Si es menor que dicho valor lo almacena en ella, sino sigue adelante (los cableados aparecen en la imagen un poco revueltos debido a la organización automática que les da NXT-G).



A continuación compara con el contenido de la variable **valorMAX** y si es mayor almacena la lectura en la variable (paso 3c).



Una vez que se registren los dos contenidos de las variables en el archivo y se cierre, esta primera parte del proceso se puede dar como finalizada.

Esto mismo lo podríamos hacer con más sensores conectados a otros puertos NXT, crearíamos un nuevo archivo de calibración para cada sensor que deseemos calibrar de un modo diferente. Para el programa que va a hacer uso de estas calibraciones he tomado como regla para los nombres de archivo **Calibra1** para el sensor conectado al puerto 1, **Calibra2** para el segundo...

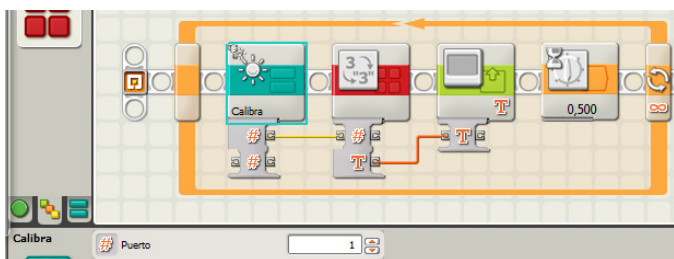
Uso del archivo de calibración

Vamos a ver primero cómo vamos a utilizar los valores registrados en el archivo. Tenemos almacenados los valores máximo y mínimo de las lecturas tomadas en unas condiciones específicas de iluminación. Ahora vamos a utilizar las matemáticas para que esos dos valores se conviertan uno en 0 (que se corresponderá con la lectura más oscura "negro") y el otro en 100 (la lectura de mayor luminosidad o "blanco").

Para llegar a ello tenemos que dar dos pasos:

- 1- Tras medir el nivel de luminosidad con el sensor, aplicar una regla de proporcionalidad que calcule el resultado deseado con un rango entre 0 y 100.
- 2- El valor directo de luminosidad que provee el sensor está comprendido entre 0 y 1023 (dado que al crear el archivo hemos utilizado este valor, aquí deberemos hacer lo mismo) pero el valor 0 se corresponde a alta luminosidad (blanco) y 1023 a ausencia de luz (negro). Si queremos convertirlo al estándar que utiliza NXT-G (de 0 oscuro a 100 brillante) deberemos restar el valor anterior de 100.

Veamos un programa ejemplo que sirve para testar todo lo anterior, y que puede ser reciclado para cualquier otra aplicación. El programa, véase la siguiente figura, lo que hará será mostrar en pantalla el valor calibrado de una lectura en tiempo real.



En el programa de la imagen, el nuevo bloque denominado **Calibra** lee el sensor conectado al puerto 1 y devuelve el valor calibrado. El resto del código es para mostrar de modo continuo las lecturas en la pantalla del NXT. Así que vamos a ver cómo es por dentro el nuevo bloque **Calibra**.

Este bloque tiene como entrada un número que se corresponde al puerto al que está conectado el sensor que

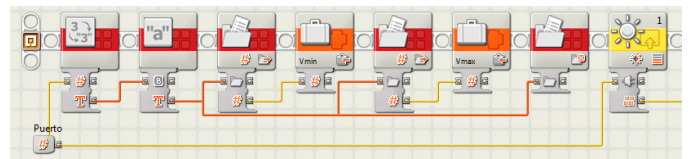
queremos leer. Si hemos nombrado los archivos del modo anteriormente mencionado, podrá devolver el valor calibrado de sensores conectados en diferentes puertos con diferentes calibraciones.

El algoritmo correspondiente a este bloque es el siguiente:

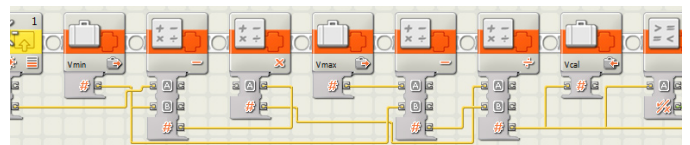
- 1- Crea el nombre de archivo a leer por medio de una operación de texto: convierte el número de entrada en texto y se lo suma a la palabra **Calibra** de tal manera que si el sensor es el 1 el archivo a leer será **Calibra1**.
- 2- Lee el primer dato del archivo y lo almacena en la variable **Vmin** previamente creada.
- 3- Lee el segundo dato del archivo y lo almacena en la variable **Vmax** previamente creada.
- 4- Cierra el archivo.
- 5- Lee el sensor de luz conectado al puerto correspondiente al número de entrada (Valor directo)..
- 6- Convierte la lectura en un valor calibrado entre 0 y 100 para almacenarlo en la variable Vcal, para lo que aplica la siguiente fórmula:

$$Vcal = [(Lectura - Vmin) / (Vmax - Vmin)] * 100$$

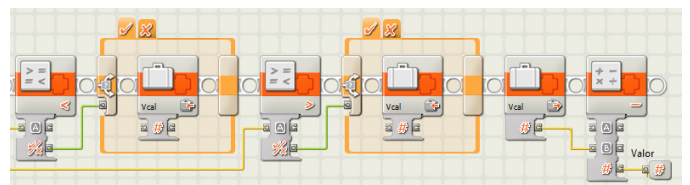
- 7- Comprueba si el resultado es inferior a 0, en cuyo caso lo convierte en 0.
- 8- Comprueba si el resultado es superior a 100, en cuyo caso lo convierte en 100.
- 9- Resta de 100 el valor anterior para conseguir el valor de salida.



En esta primera imagen podemos ver los 5 primeros pasos.



A continuación se hace el cálculo.



Para acabar comprobando que el resultado está dentro del rango deseado y normalizando el resultado.

Nota final

A partir de aquí cada uno ha de adaptarlo a sus necesidades, diferentes calibraciones para diferentes sensores o diferentes calibraciones para un mismo sensor. Esto supondrá que serán necesarios tantos archivos como diferentes calibraciones se necesiten.

#



Lrobotikas.net

Robótica Educativa y Recreativa