

Iniciación a la robótica con LEGO® MINDSTORMS, 13ª entrega

Programa el NXT con LabVIEW

Por Koldo Olaskoaga

Llega la FTC

Este año se celebra por primera vez un torneo de la FTC[1] en España. En este estreno, que tendrá un carácter internacional, participarán equipos de Barcelona, Jaén, Gipuzkoa, Madrid y Málaga acompañados por equipos de Rusia, Holanda y EUA (en tele-competición). El torneo se celebrará en Tarragona los días 16 y 17 de marzo.

El hardware en la FTC es una combinación de LEGO MINDSTORMS y TETRIX[2]. El robot ha de ser controlado obligatoriamente con el NXT (parte del tiempo con funcionamiento autónomo y otra parte teledirigido) y puede ser programado en LabVIEW para LEGO MINDSTORMS o en RobotC.

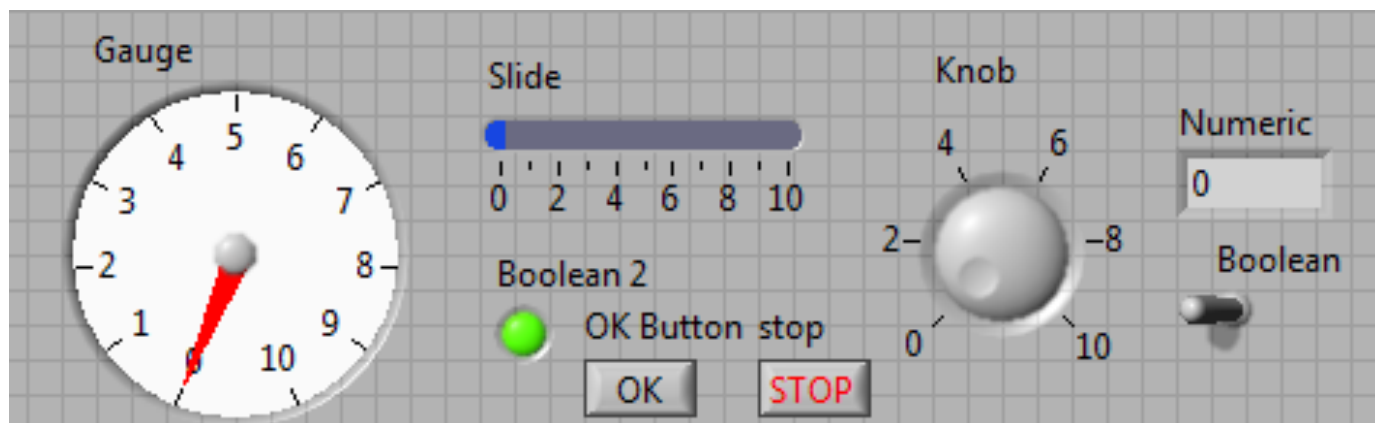
LabVIEW

LabVIEW es la plataforma y entorno de desarrollo para diseñar **sistemas**. NXT-G está creado sobre LabVIEW. Utiliza un lenguaje de **programación** visual gráfico.

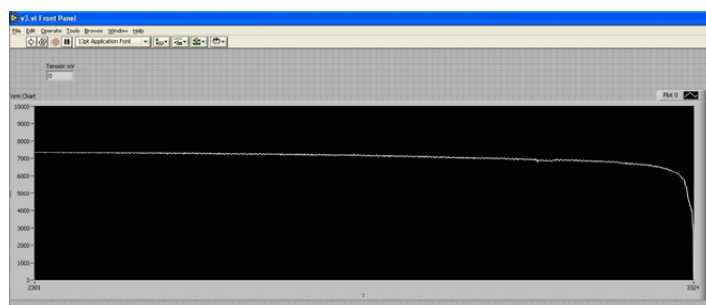
Es un entorno de programación comercial de uso relativamente intuitivo. Este artículo es continuación del que pudiste leer en el número anterior de HBM. Aquí quiero mostrar las líneas generales de programación con LabVIEW tomando como referencia el mismo proyecto. No voy a repetir lo que está dicho en el primero sino mostrar cómo se desarrolla en LabVIEW.

Edición básica de programas

Un programa de LabVIEW consta de dos espacios relacionados: el panel frontal y el diagrama de bloques (este último contiene al programa propiamente dicho). El panel frontal se puede utilizar para monitorizar sensores, motores, mostrar valores resultado de cálculos internos o incluso modificar parámetros de funcionamiento. ¿Quieres montar un pequeño invernadero controlado por el NXT? El panel frontal puede ser el panel de control de todo el sistema. En la siguiente imagen se pueden ver algunos de los controles disponibles.



El panel frontal también puede ser muy útil en proyectos educativos de recogida y representación sistemática de datos. Es algo que ya ofrece la versión educativa de NXT-G, aunque el potencial de LabVIEW es muy superior. En la imagen siguiente puede verse un ejemplo en el que se muestra el panel frontal de un programa cuyo objetivo es analizar el proceso de descarga de la batería del NXT[3].



[1] FTC: <http://goo.gl/WNonY>

[2] TETRIX: <http://www.tetrixrobotics.com/>

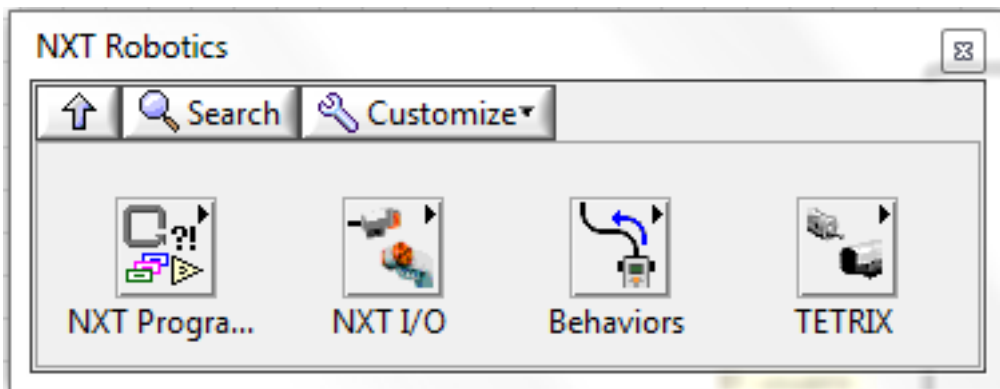
[3] Ver proyecto completo en <http://goo.gl/pFUoG>

El segundo espacio de trabajo es el diagrama de bloques sobre el que se desarrolla el programa que definirá el comportamiento del robot. Si lo que se desea crear es un programa que se ejecute en el NXT, será el único de los dos espacios que utilizaremos.

En la imagen siguiente puede verse el fragmento de programa que muestra en pantalla la información e instrucciones de inicio.



Los bloques de programación se obtienen de la paleta de funciones y se unen por medio de cables para definir el flujo del programa. Cada bloque está configurado con unos valores por defecto que habrá que modificar siempre que sea necesario. En la imagen anterior los valores numéricos en azul representan la línea en la que ha de representarse el texto en la pantalla del NXT.



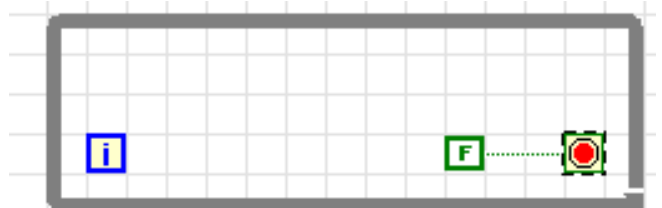
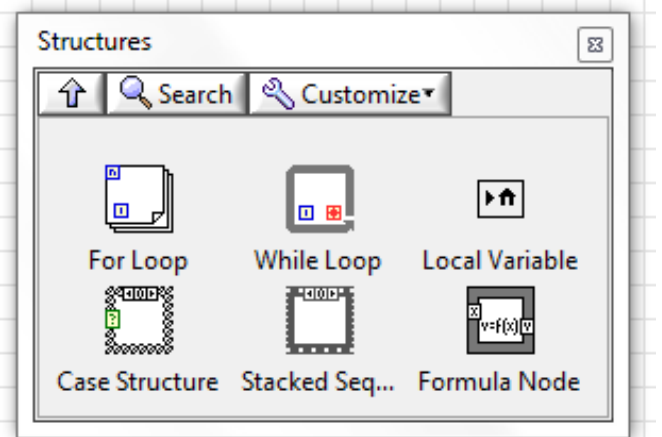
Los programas pueden ejecutarse tanto en el NXT como en el PC. En este segundo caso se puede utilizar todo el potencial de LabVIEW controlando desde el PC los recursos del NXT.

Estructuras de control

LabVIEW ofrece varias estructuras tal y como puede observarse en la imagen siguiente. Si buscamos algún paralelismo con NXT-G, la estructura While Loop se corresponde al bloque Bucle y la estructura Case al bloque Bifurcación.

Además de estas dos estructuras también se puede utilizar la estructura For Loop, común en otros lenguajes de programación pero sin una equivalencia directa en NXT-G y Stacked Sequence, una estructura propia de LabVIEW. Veamos primero algún ejemplo de cómo se ha utilizado la estructura While Loop en este proyecto. Esta es la estructura más utilizada en el programa, ya que es necesaria en todos aquellos casos en que haya de repetirse un fragmento del programa durante un número indeterminado de veces. El programa finalizará el bucle en el momento que se cumpla la condición que corresponda en cada caso. En la imagen siguiente puede verse un bucle que se repetirá de modo indefinido. El círculo rojo de la derecha expresa que el bucle finalizará cuando reciba un "Verdadero", algo que no sucederá nunca en este caso.

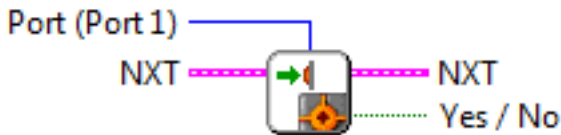
Una de las tareas para la que ha sido necesario utilizar esta estructura es para esperar a que una u otra tecla sea pulsada.



De hecho, hay que esperar a que uno de los tres sensores de contacto o el botón naranja sea pulsado. Para ello se ha utilizado el bucle de la derecha:

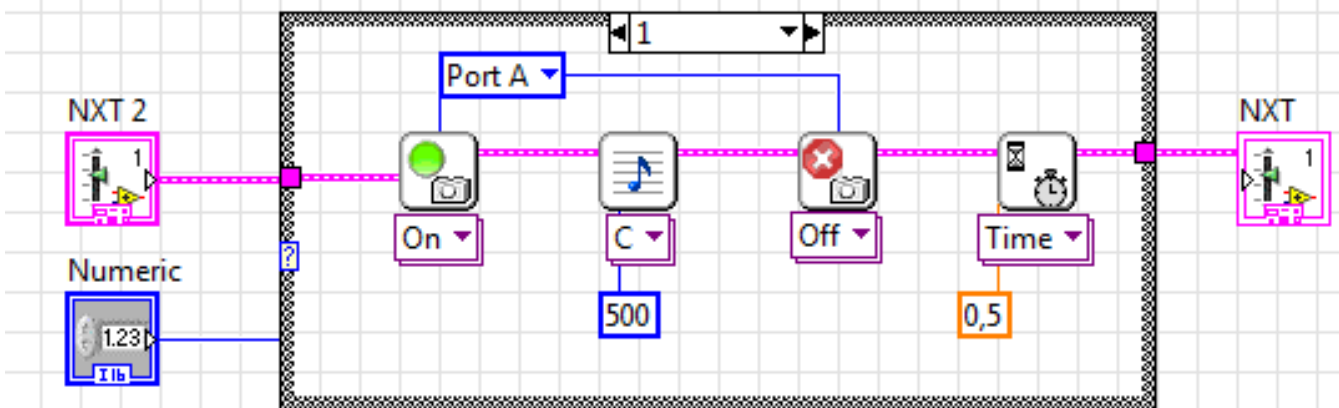
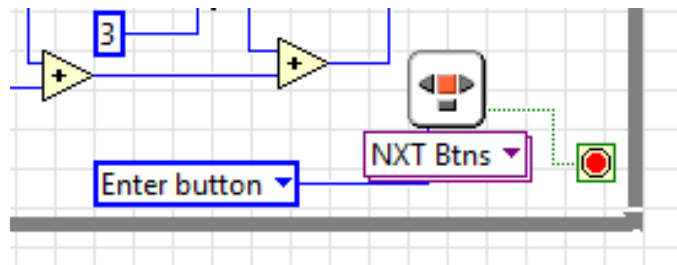
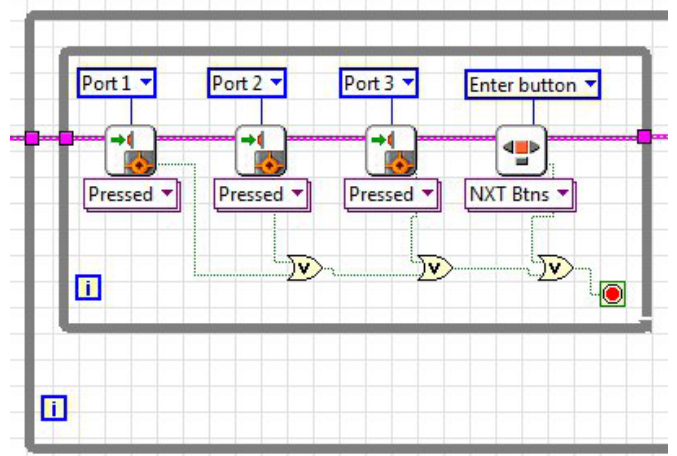
En este bucle el programa lee una y otra vez los 4 sensores. Para determinar si alguno de ellos ha sido pulsado utiliza la operación booleana OR, de tal manera que si alguno de los sensores es pulsado el resultado de la operación será "Verdadero" y el programa continuará. En la imagen siguiente pueden verse los conectores disponibles en los bloques Leer Sensor.

Read Touch (Pressed) [Touch_Pressed.vi]



La condición a cumplir también puede representarse de un modo independiente al flujo principal del programa, tal y como se hace para acabar con una de las fases del programa por medio del botón naranja del NXT.

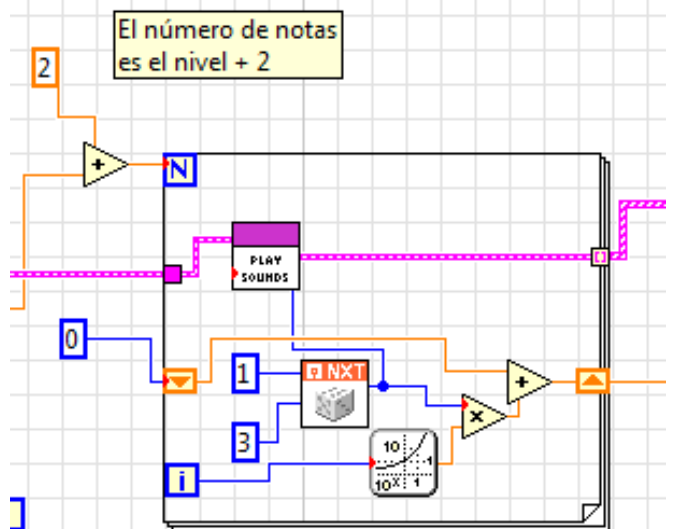
A la hora de tomar decisiones se ha utilizado la estructura Case, equivalente a la estructura Bifurcación de NXT-G. Entre otras cosas se ha utilizado para reproducir el sonido y encender la luz correspondiente a cada tecla. Esto se ha hecho utilizando un nuevo bloque de programación (un nuevo SubVI en la terminología de LabVIEW). Es un bloque que tiene como entrada un valor numérico entre 1 y 3 que al desplegarlo presenta el siguiente aspecto:



La morfología de esta estructura es similar a la del bloque Bifurcación cuando no está seleccionada la casilla de verificación Vista Plana. En este caso se puede ver el código correspondiente al valor de entrada 1.

La tercera estructura utilizada en el programa ha sido For Loop. Una de sus aplicaciones ha sido generar la secuencia aleatoria a reproducir. Para ello ha sido necesario generar tantos valores aleatorios entre 1 y 3 como el número que representa el nivel +2. Se puede ver en la parte superior izquierda la entrada de dicho valor.

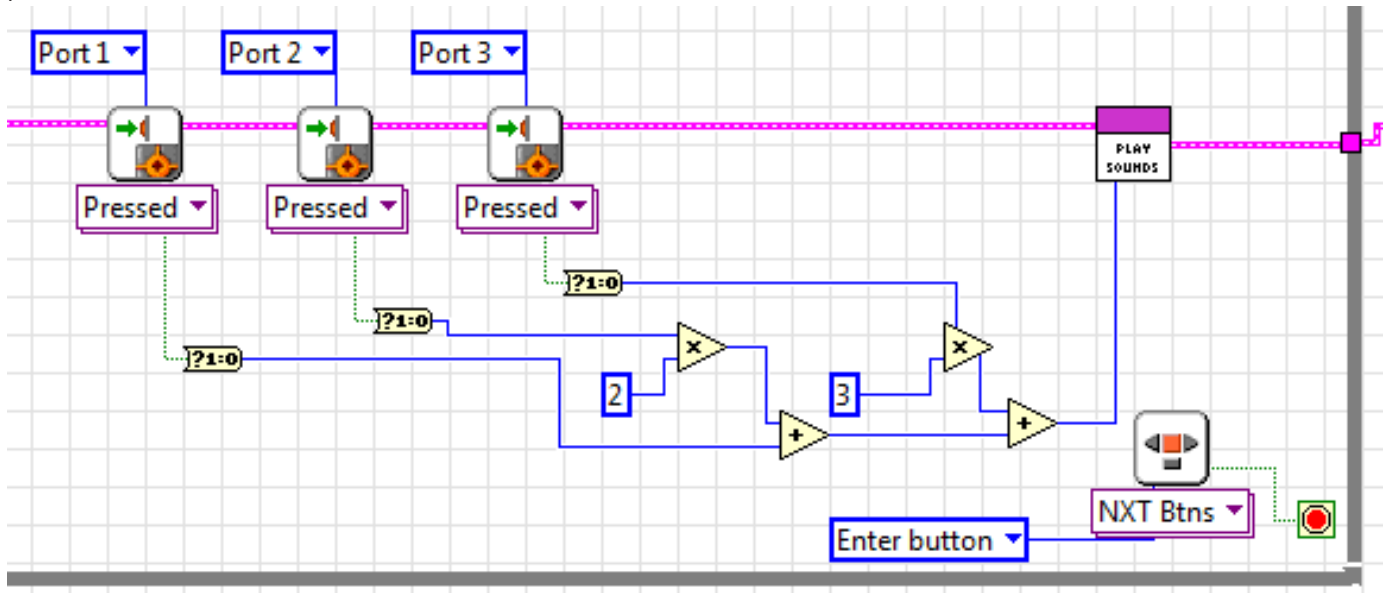
Aunque en LabVIEW la secuencia se puede almacenar como una matriz, se ha utilizado el mismo subterfugio que con NXT-G para convertir la secuencia en un valor numérico, multiplicando el primer valor por 1, el segundo por 10... y sumando todos ellos. Para ello se utiliza la función exponencial 10x que obtiene el ordinal (con valores comprendidos entre 0 y n-1) correspondiente al bucle actual desde la terminal (i).



Aunque no se haya mencionado anteriormente, la estructura While Loop también permite utilizar el número de bucles desde su terminal (i).

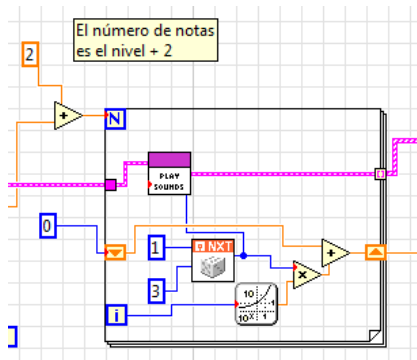
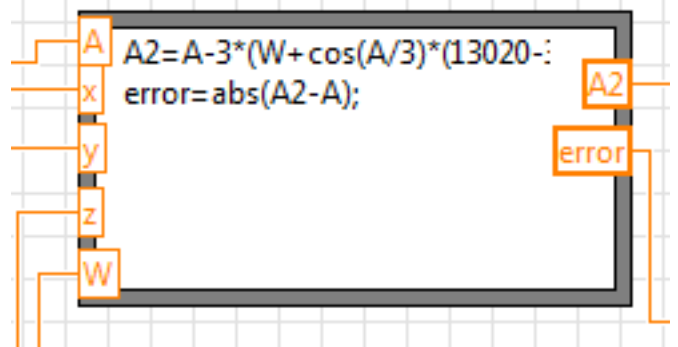
Datos y operadores

En LabVIEW los datos son transportados por medio de cables, del mismo modo que se hace en NXT-G. Las operaciones se ejecutan una vez que todos los datos necesarios para ellas estén disponibles. LabVIEW dispone de un importante conjunto de operadores para datos de tipo numérico, booleanos, cadenas de texto, matrices... En la operación representada en la siguiente imagen primero se convierte el Verdadero / Falso de los sensores en un valor numérico 1/0 antes de utilizar los operadores aritméticos suma y multiplicación para determinar qué sensor ha sido pulsado.



Además de utilizar estos operadores en modo gráfico, como se ha hecho en este proyecto, también es posible introducir expresiones matemáticas vía la estructura Fórmula Node. Esta estructura evalúa fórmulas matemáticas y expresiones en un formato similar a C.

A diferencia de lo que se hace en NXT-G, en este programa no se ha creado ninguna variable en la que almacenar datos y de la que leerlos cuando sea necesario. Para ello se utiliza el cableado. Un ejemplo de ello puede verse en la estructura For Loop del punto anterior.



Los dos rectángulos con los triángulos naranjas son los elementos que permiten convertir un valor de salida de una de las iteraciones de la estructura y utilizarla como entrada en la siguiente iteración. En la primera iteración se ha inicializado con el valor 0 y en cada iteración se le ha sumado el resultado de la operación anteriormente explicada. Puede observarse que la salida del bloque número aleatorio está unido al SubVI Play Sounds para reproducir el sonido correspondiente.

Comentario final

LabVIEW es un software comercial con un precio de unos 150€ para licencias individuales y precios más reducidos para licencias de varios puestos o licencias de centro. Esto es algo que limita su uso al ámbito educativo. Si lo que se desea es monitorizar experimentos o controlar un sistema diseñado con LEGO® MINDSTORMS, el panel frontal y su capacidad de proceso de datos, ofrecen una herramienta muy potente e interesante.

#

