An introduction to Robotics with LEGO® MINDSTORMS (XIII)

Programming the NXT with LabVIEW

By Koldo Olaskoaga

FTC is here

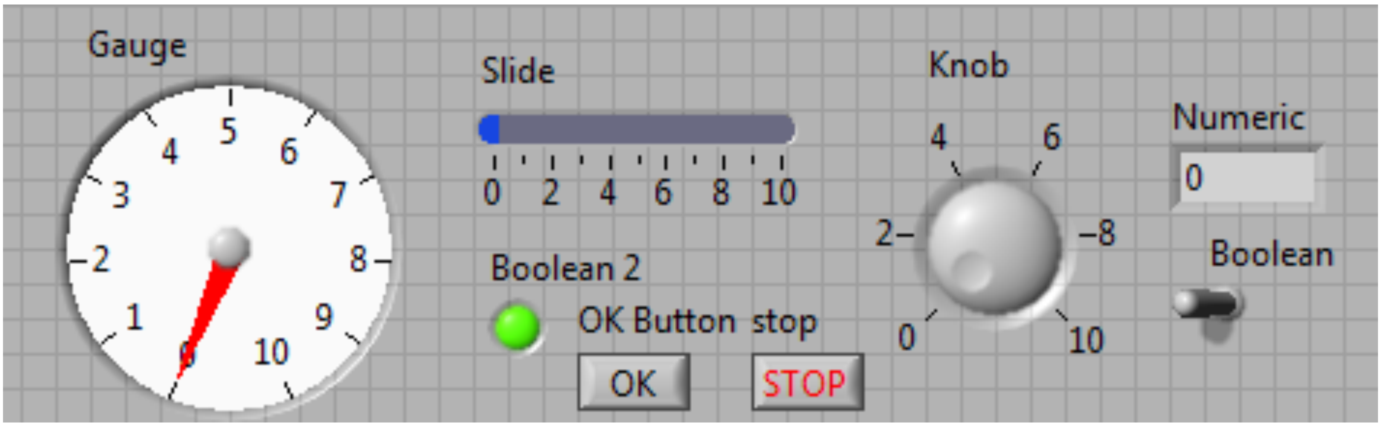
This year the first FTC[1] tournament is celebrated in Spain. In this premiere, which has an international character, teams from Barcelona, Jaén, Gipuzkoa, Madrid and Malaga will compete alongside teams from Russia, Holland and the USA (via tele-competition). The tournament will take place in Tarragona on march 16 and 17.
The hardware used in FTC is a combination of LEGO® MINDSTORMS and TETRIX[2]. The robot must be controlled by the NXT (part of the time autonomously and part of the time remote controlled) and can be programmed using LabVIEW for LEGO MINDSTORMS o RobotC.

LabVIEW

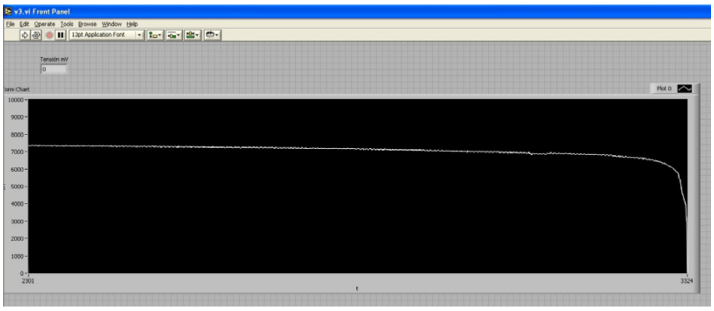
LabVIEW is a plataforma and environment for designing systems. NXT-G has been developed in LabVIEW. It uses a visual programming language.
It is a commercial programming environment that is quite intuitive. This article is a continuation of the one from the previous edition of HBM. In this one I would like to show you the basics of programming with LabVIEW, using the same project as a reference. I am not going to repeat everything that was said in the first article, but rather show how to develop it in LabVIEW.

Basic program editing

A LabVIEW program consists of two related spaces: the front panel and the block diagram (this last one contains the program itself). The front panel can be used to monitor sensors, motors, show values from internal calculations or even modify parameters of the program. Would you like to build a small greenhouse and control it with the NXT? he front panel can be the control panel for the system. In the following image you can see some of the available controls.



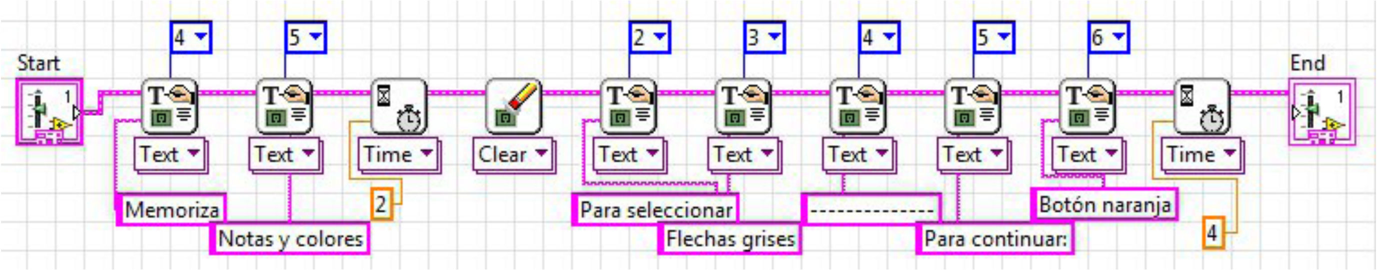
The front panel can also be very useful in educational projects which involve data collection and representation. This is something the educational version of NXT-G already includes, although the potential of LabVIEW is much bigger. In the following image you can see an example that shows the front panel in a program that analyses the discharge process of an NXT battery[3].



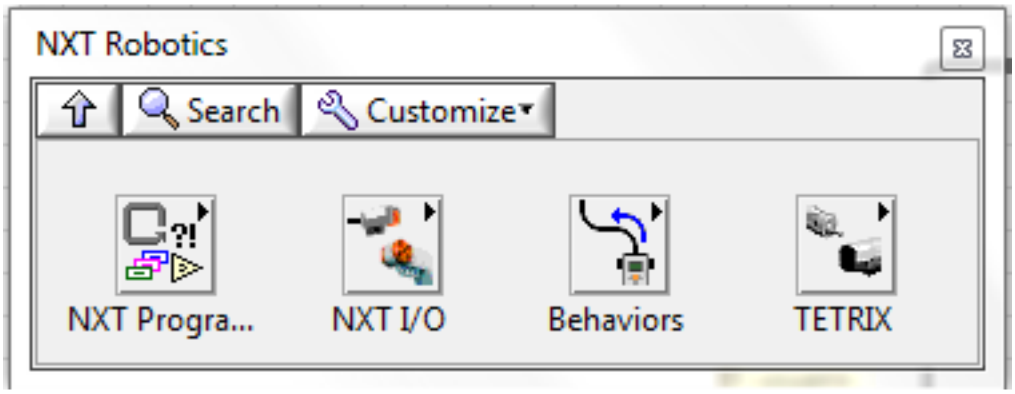
[1] FTC: http://goo.gl/WNonY
[2] TETRIX: http://www.tetrixrobotics.com/
[3] See the complete project at http://goo.gl/pFUoG

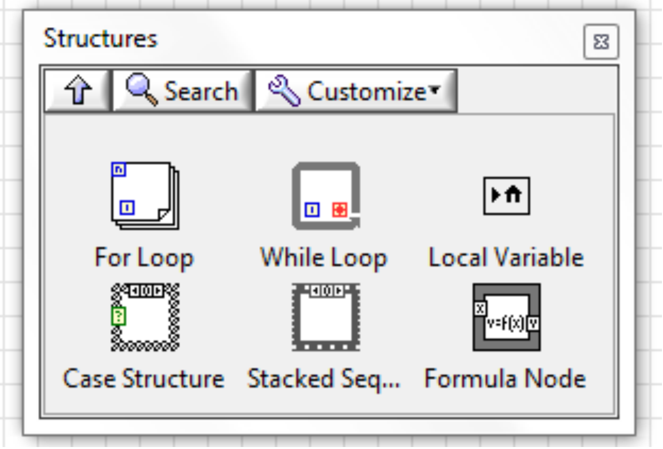
The second workspace is the block diagram in which the program that defines the behaviour of the robot is developed. If what you want to create is a program that runs on the NXT, it will be the only space you use.

In the following image you can see a fragment of the program, that shows the information screen and starting instructions.



The programming blocks are taken from the functions palette and are connected by means of wires that define the program flow. Each block is configured with a set of default values that need to be modified as needed. In the previous image the numerical values in blue represent the line on which the text should be displayed on the screen of the NXT.



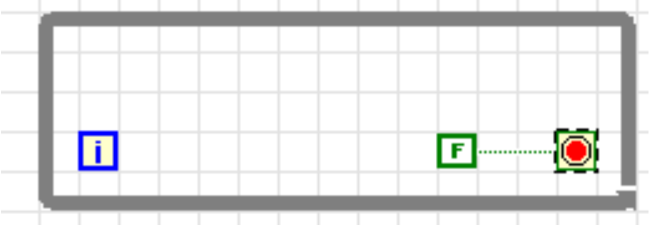


Programs can be run both on the NXT and on the PC. In the second case all of the potential of LabVIEW can be used, controlling the NXT resources from the PC.

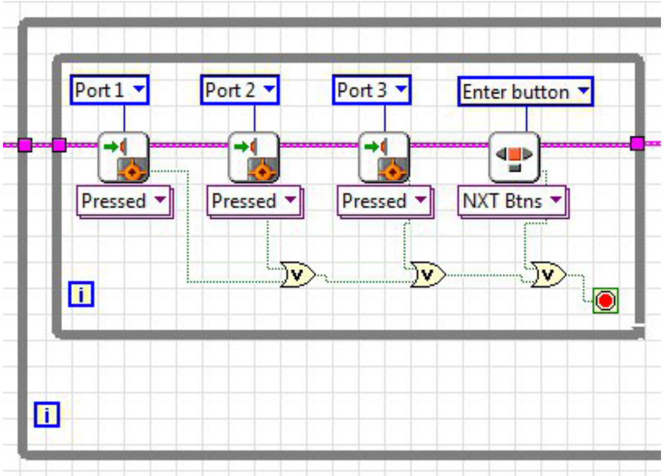
Control structures

LabVIEW offers several structures as can be seen in the following image. If you look for parallels with NXT-G, the While loop corresponds to the Loop block and the Case structure to the Switch.

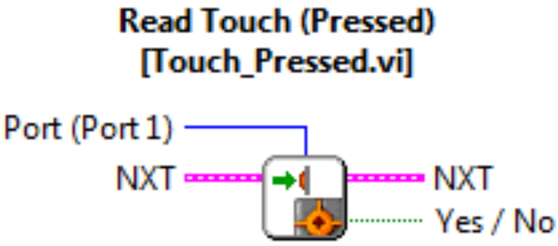
In addition to these two structures you can also use the For Loop, common n other programming languages, but without a direct equivalence in NXT-G and Stacked Sequence, a LabVIEW specific structure.
Lets first see an example of how the While Loop has been used in this project. This structure is the most used one in the program, as it is necessary in all the cases where a fragment of the program needs to be repeated a specific number of times. The program exits the loop the moment the corresponding condition is met. In the following image you can see a loop that will repeat itself indefinitely. The red circle on the right shows the loop will end when it receives a “True” signal, something that will never happen in this case.

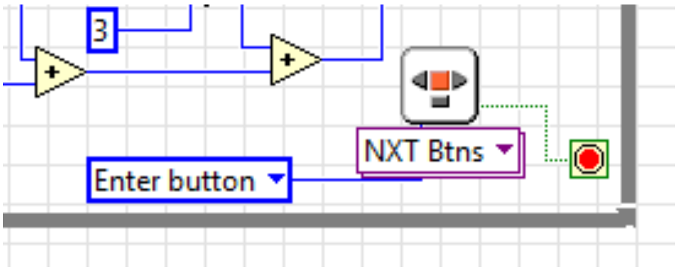


One of the tasks for which it has been used is to wait for one of the keys to be pressed. As a matter of fact, it needs to wait for one of the three contact sensor or the orange NXT button to be pressed. To this end the loop on the right has been used.



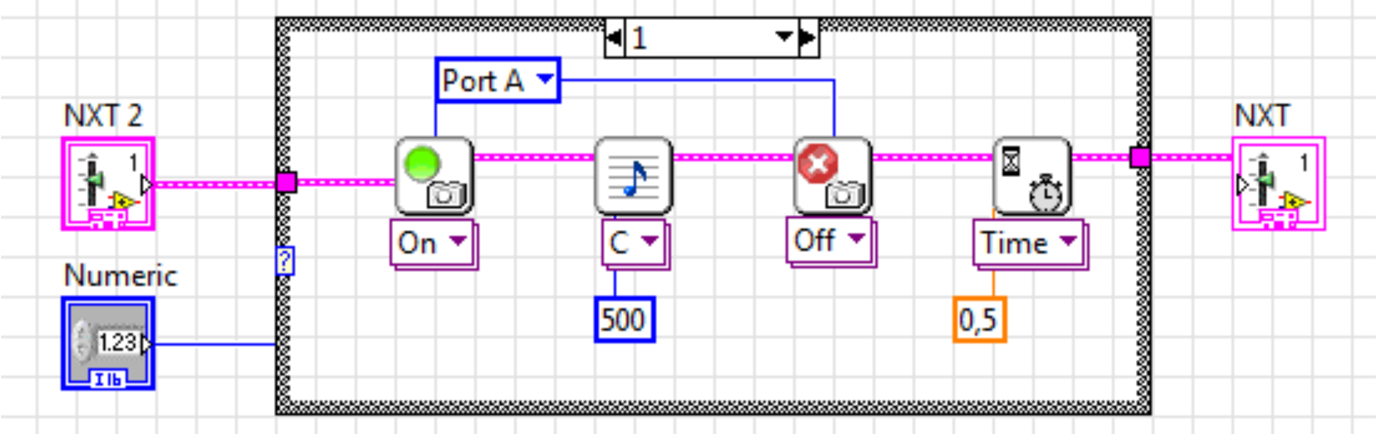
In this loop, the program reads the 4 sensors time after time. To determine if any of them has been pressed it uses the boolean OR, so that if any of the sensors is pressed the result of the operation will be "true" and the program will move on. In the following image you can see the connectors that are available on the Read Sensor block.

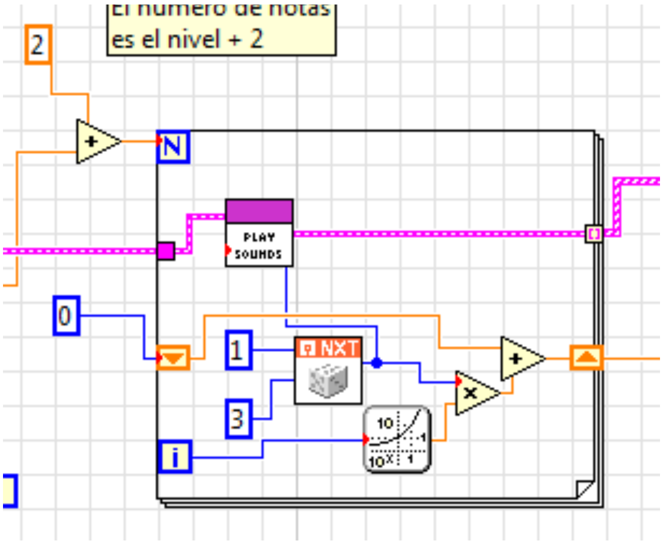




The condition that must be met can also be represented independently at the start of the program, just like is done to end one of the phases of the program by means of the orange NXT button.

For taking decisions the Case structure has been used, equivalent to the NXT-G Switch. Among other things it has been used to reproduce a sound and switch on the corresponding light to each key. This has been done using a new programming block (a new SubVI as it is called LabVIEW). This block has a numeric value between 1 and 3 as an entry and looks as follows when opened:





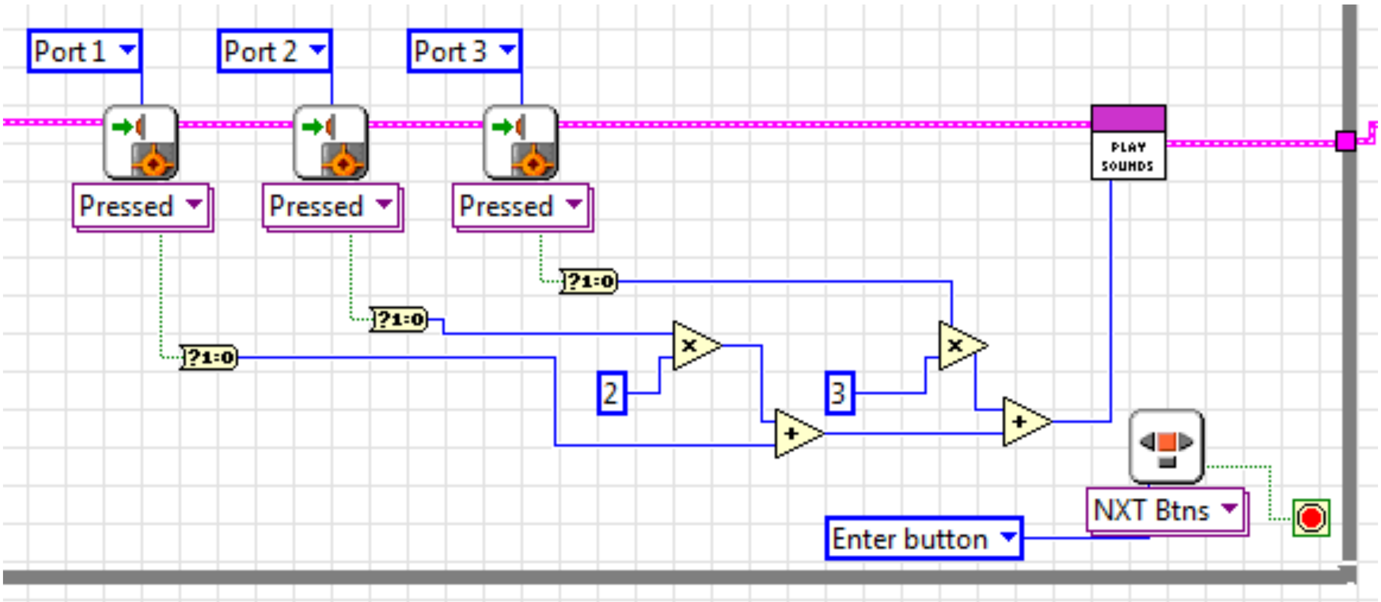
The morphology of this structure is similar to the switch block when the check box for flat view is not selected . In this case you can see the code corresponding to the input value 1.
The third structure used in the program has been the For Loop. One application is to generate the random playback sequence. To this end it has been necessary to generate many random values between 1 and 3 as the number representing the +2 level. You can see the input value in the upper left corner.

Although in LabVIEW the sequence can be stored in a matrix, the same subterfuge as in NXT-G has been used to convert the sequence in a numeric value, multiplying the first value by 1, the second by 10... and adding them all. For this purpose the exponential function 10x is used to obtain the ordinary number (with values between 0 and n-1) corresponding to the current loop from the terminal (i). Although it has not been mentioned before, the While Loop structure also allows the use of the number of loops from the terminal (i).

Data and operators

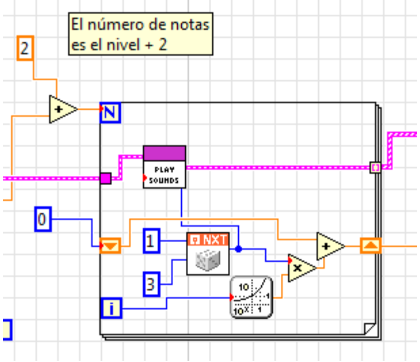
In LabVIEW, data is transported by wires in the same way as in NXT-G. The operations are carried out when all the necessary data is available.

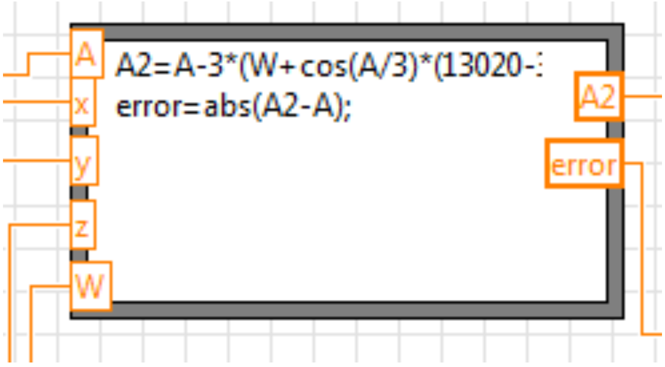
LabVIEW has an important number of operators for data of different types: numerical, boolean, text chains, matrices, etc. In the operation that appears in the following image, the True/False input from the sensor is converted into the numerical value 1/0 before using the arithmetic addition and multiplication operators to determine which sensor has been pressed.



In addition to these operators in graphic mode, as has been done in this project, it is also possible to introduce mathematical expressions through the Formula Node structure. This structure evaluates mathematical formulas and expressions in a format that is similar to C.

Contrary to what is done in NXT-G, in this program no variables have been created to store data and from which to read when necessary. For this purpose the wires have been used. An example can be seen in the For Loop of the previous point.





```
A2=A-3*(W+cos(A/3)*(13020-:
error=abs(A2-A);
```

The two rectangles with orange triangles are the elements that allow the conversion of an exit value of one of the iterations of the structure to be used as en entry for the next iteration. In the first iteration the value has been initialised to 0 and in each subsequent iteration the result of the previous operation has been added as previously explained. As can be seen, the exit of the random number block is connected to the SubVI Play Sounds in order to reproduce the corresponding sound.

Final remarks

LabVIEW is a commercial software with a cost of about €150 for an individual license and better prices for several workstations or licences for schools. This is something that limits its use to the educational environment.
If what we want to do is monitor experiments or controlling a system designed with LEGO® MINDSTORMS, the front panel and its capability to process data offer a powerful and interesting tool.
#