

Homemade sensors for NXT: a multiplexor

By *NXTorm*

As you all know, the NXT is the programmable brain of the current LEGO® MINDSTORMS universe. You can connect official sensors to it, but also your own sensors. This article is an introduction into the DIY of MINDSTORMS: you will learn to program your own multiplexed touch sensor. Come again, a what? Well, it's a number of touch sensors that are connected to a single NXT sensor port. Let's get going!

WARNING

If you don't want to break anything, stop reading. I won't be held responsible if anything happens to your NXT. But it will certainly be fun. More information: www.nxtorm.es

NOTES:

All the experiments described below have been tested previously.

In order to understand this article it is helpful if you are familiar with the basic programming functions of NXT-G. Even so, I will explain everything step by step.

MATERIALS

The materials we will use are cheap, relatively easy to obtain and won't require soldering. You will find them in electronics shops or on the Internet; there are many options. With this in mind, let's get to the list of materials and their pictures.

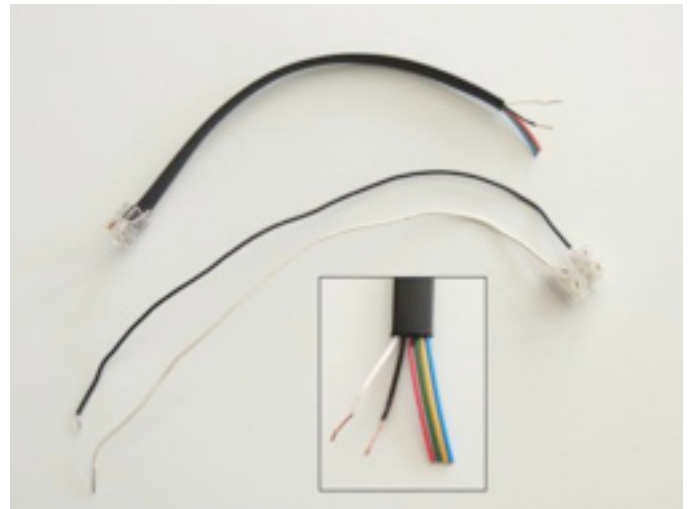
1. An original LEGO MINDSTORMS connection cable that we will cut in half.
2. A test modulo or "breadboard" (see image).
3. Four push buttons that can be stuck into the breadboard.
4. Black and white wire for connections, both long ones and short ones for connections.
5. Three resistors of ¼ watt (1/4W) of 2K2, 4K7 and 6K8. The colours are: 2K2: red-red-red; 4K7: yellow-violet-red;



- 6K8: blue-grey-red.
6. Two screw terminals for the cables.

THE CABLES

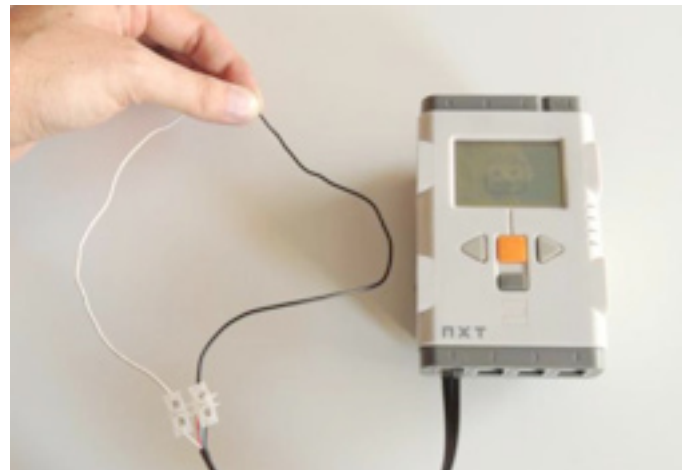
After cutting the original NXT cable in half, strip only the tips of the black and white cables to leave the inner cable bare. The rest you can leave unstripped.



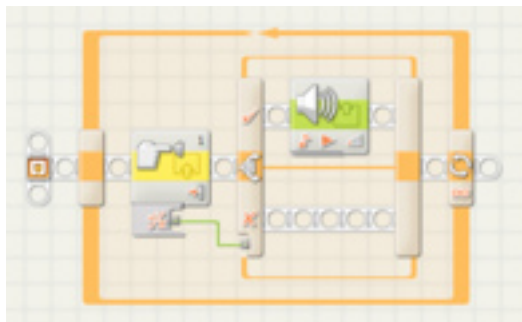
It is easiest to fit them to a screw terminal and connect them the stripped end of a rigid cable (wire) as in the picture. These are easier to connect to the breadboard than the original NXT cable, which is made up of several thin copper wires and so less rigid.

FIRST TEST

The easiest thing is to start the NXT, go to the Try-me menu and select Try-Touch, i.e. the touch sensor. Connect the cable to port 1, look at the screen and let the stripped ends of the black and white cables touch.



The first time I did this test I was really excited, but of course it is a little rudimentary. We can make a small improvement with the following program:



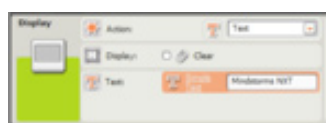
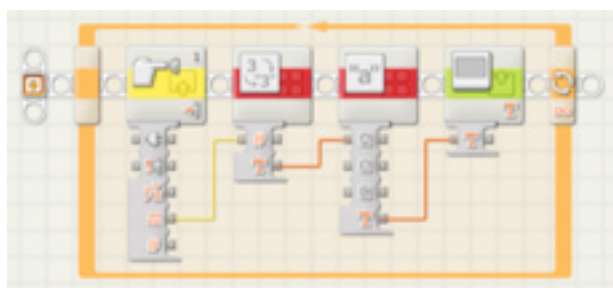
Configuration of the program

Contact sensor: select "Pressed".
 Sound Action: "Tone"; Note: for 0.1 sec; Wait for Completion: No;
 Switch: Select "Value" and "Logic" (see image).
 Loop: unlimited

As an aside, you can select the icon of the Touch Sensor Block on your computer and then check the lower left corner of your screen as you connect and disconnect the cable.

OBTAINING "Raw Value"

Although there are several ways to obtain data, we will concentrate on the RAW value the touch sensor provides, and which goes from 0 to 1023. Click below the block to access the hub as seen in the image and write this program. The details are below the image.



In the configuration panel of the Text block, write several spaces in the box marked "B". The Display block is configured for Text and the check box for Clear is unchecked. This program prevents the screen from flickering and digits from previous numbers showing up.

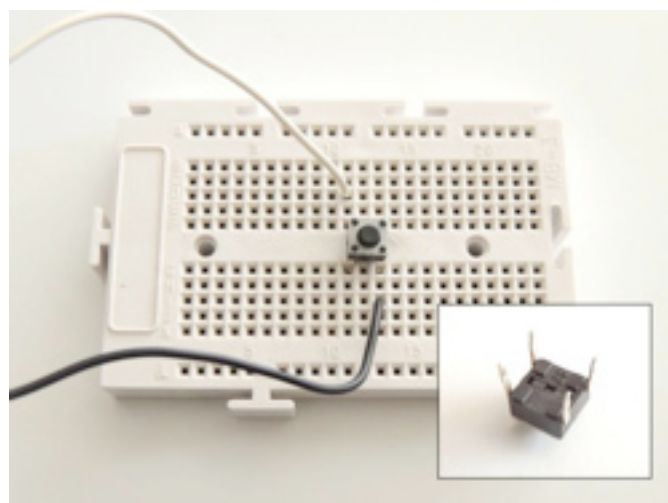
THE BREADBOARD

There are many models, but they all have a common row at the top and at the bottom, i.e. a row in which all the pins are connected. There are also columns of 5 connected pins. Are you in doubt if two pins are connected? Use the NXT with the previous program!

THE FIRST PUSH BUTTON

Push buttons have 4 pins to fit them into the breadboard. The pins are internally connected in pairs. How to tell them apart? Connect them diagonally (see image or... use the NXT to find out which pins are connected!

How? With the push button on the breadboard, connect the black and white cables randomly and use the previous program. Do you get a 0? In that case the push button is incorrectly connected: change the cables until you get a reading of 1023. Then the connection will be right.



ONE MORE STEP: A RESISTOR

Now we are going to add a resistor of 2K2. The white and black cables are connected to a common row (one top, one bottom) and I have used a small white cable for the connection. With the 2K2 resistor the value is approximately 181, because not all resistors are the same.



A SECOND PUSH BUTTON WITH RESISTOR THE IMPROVED PROGRAM

Now that we have a first push button, let's add another one together with a 4K7 resistor. Simply duplicate the earlier construction.



What value do you get when you push the second resistor? In my case it is 322. As you can see, THE RAW VALUE DEPENDS ON THE EXACT VALUE OF THE RESISTOR YOU USE. You can use this idea for many uses, including a multiplexor or a resistance reader.

THE MULTIPLEXOR

To finish our multiplexor we simply need to add another two push buttons. With one of them we will use the 6K8 resistor and the last one will go without OK resistor, i.e. a bit of wire. We already know the RAW value without a resistor: 0. My RAW value for the 6K8 resistor is 409.



So now, simply reading the value on the screen you can know which push button is pressed. We have built 4 home-made touch sensors using a single NXT port. And you could add even more!

Looking at the screen you know which push button is pressed, but can the NXT do the same? Of course! But the program gets a little more complicated. Among other things because sometimes values can vary between two number, e.g. 409 and 410. The "dirty" way of fixing this is to add more tabs for these values. Write down the values for your buttons on a piece of paper and we can continue:



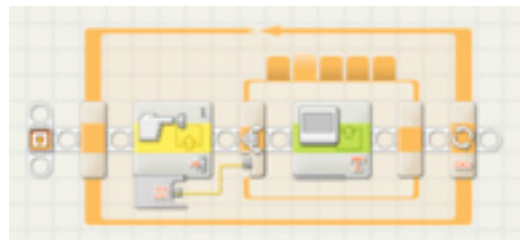
In the configuration panel of the Switch block, select "Value" and "Number" (see image). Now deactivate the Flat View check box.

On the right in the configuration panel of the Switch block you will see three buttons. +, - and *. The + serves to add more tabs, up to 5 in our case, corresponding to each one of the push buttons plus one for 1023 when no button is pushed.

Make 1023 the default with the *. Remember my numbers are: 0, 181, 323, 409 and 1023, which are written in the window below.

Now you'll need to select each of the tabs with the mouse to program each one for the action you want to associate with the button: the movement of a car (a remote with 4 buttons), showing a number on a screen, playing a sound as if it were a piano, etc.

Each tab in the Switch corresponds to the action of one push button and the additional tab is for the default case when no button is pushed.



And that's it, we're ready. I hope you can take advantage of this idea for your own ideas and applications: the possibilities are endless. And it looks like things will work exactly the same on the upcoming EV3. Good luck!

Would you like more information? www.nxtorm.es
#