

Using Data Hubs For LEGO Mindstorms NXTg

What's a Data Hub.



Click here, opens the data hub



A data hub is a more advanced feature of programming LEGO Mindstorms NXT with certain NXT programming blocks.

A data hub contains data plugs that allow the programmer to connect data wires to the plugs.

The data wires pass information (data) between the blocks and allow the blocks to process this information in a variety of ways. This data can be dynamic (changing all the time) or static (doesn't change). A block can send information (output plug) or receive information (input)

By using a combination of sensor, variable and math's programming blocks with data wires the programmer can get the robot to do a lot more by itself rather than the programmer having to put in this information. i.e. a more automatic / autonomous robot.

← INPUTS OUTPUTS →

Input plugs (receive data) are on the left of the hub, output plugs (send data) are on the right side of the hub. You plug incoming data from another hub into the **INPUT** side . On the **OUTPUT** side you send data from your hub to another hub's input plug

There are different types of data plugs depending on the block /sensor and the type of data you wish to send or receive. Data Wires pass different types of data between the blocks.

Match the colour of the wire to the matching port to pass data .

i.e. Number data wires are yellow

Logic (yes/ no) data wires are green

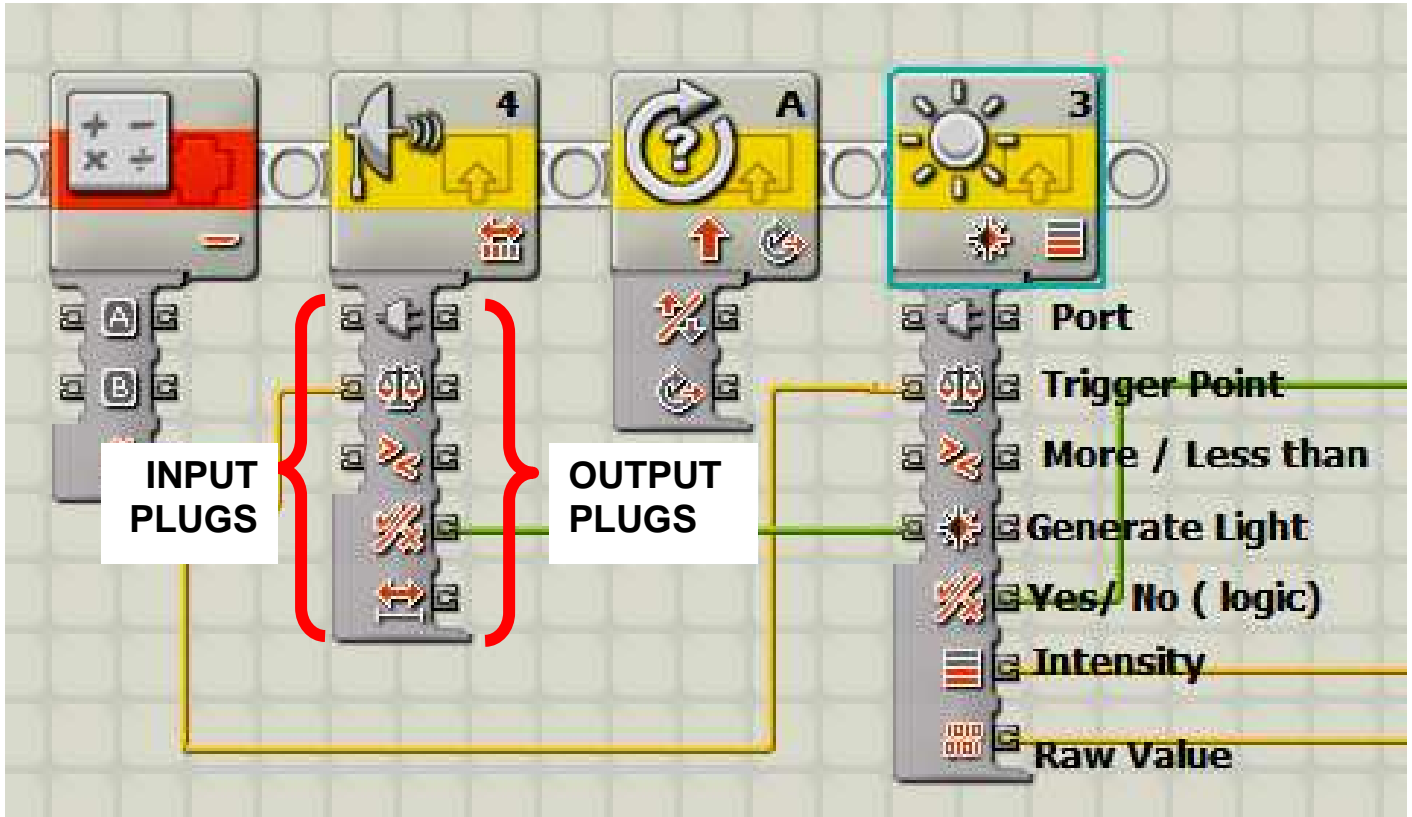
Text data wires are orange

Broken (bad / wrong) data wires are dashed grey

If you connect the wrong type of data to a port then you will get a broken / bad data wire. E.g. if you connect text to a Trigger (number) plug

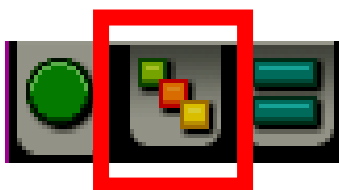
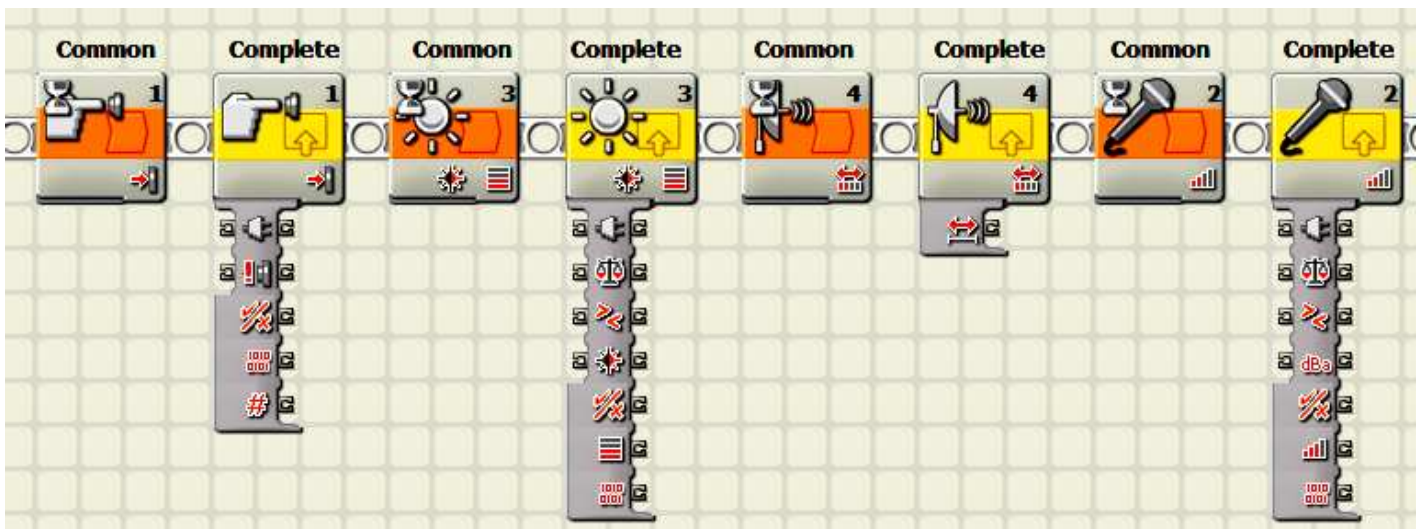
To delete a data wire left mouse click on it and then back-space. A program with broken data wires will not download into the NXT.

Some blocks with their data hubs and data wires.




Most of the blocks that have data hubs are contained in the **COMPLETE** palette.

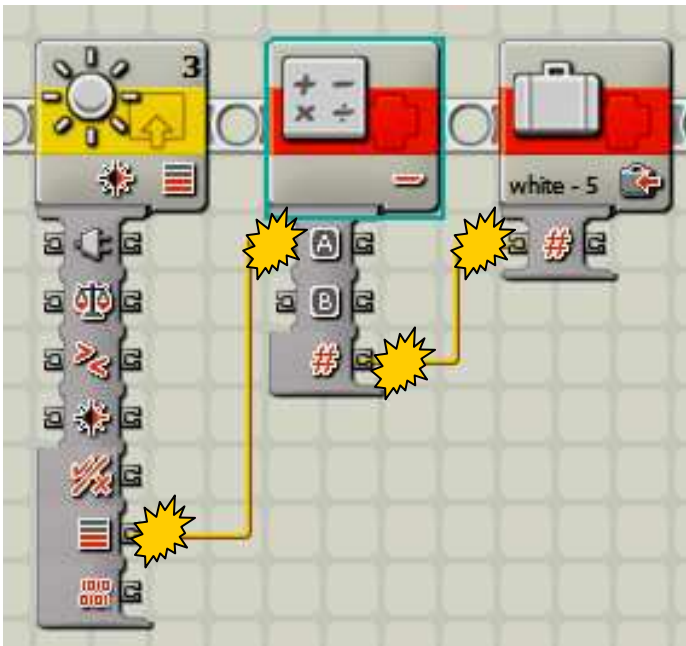
Not all programming blocks have data hubs and sometimes there is two versions of a block, one with a Data hub and one without. The basic sensor blocks in the **COMMON** palette don't have data hubs. The **SENSOR** blocks in the **COMPLETE** palette do have data hubs



COMPLETE palette



To join an output data wire to an input data plug (of the same type eg . number to number) left mouse click on the output plug and  then left mouse click on the corresponding input plug



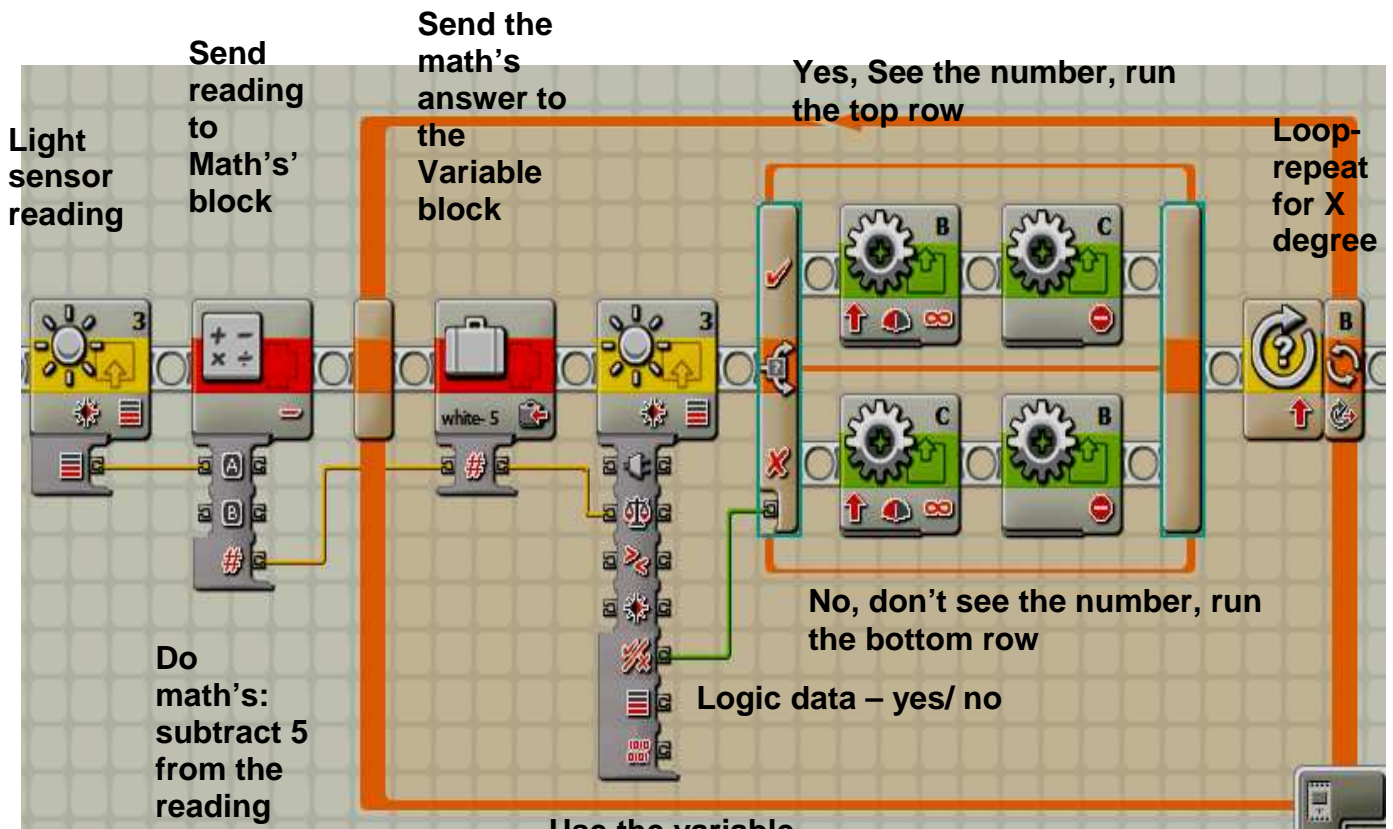
In the example at left the light sensor is sending the light intensity reading (number) to the math's block. In the math's block 5 is subtracted from the light intensity and then this answer (number) is sent to the variable block.

In most basic programs the data that is sent between blocks will be a number or logic .

The following program example shows an automatic line reader.

The light reading is taken and processed by the math's block. This number is then stored in the variable block. The number in the variable block is sent to the trigger plug on the light sensor and the light sensor then keeps comparing the trigger number with whatever light value it is

currently seeing. If the light sensor sees that same number (yes) it will run the top row of the switch block and if no, it will run the bottom row of the stitch block



Use the variable- compare with the current light sensor reading- Yes, I see the number, No I don't

Postscript:

The Math's block is an interesting block that allows the programmer to process data the block receives. In the previous example (assumes black line on white background) the robot took a light reading on the white background and sent this reading via a data wire to the Math's block. The Math's block then subtracted 5 from this reading, the program then uses this lower (darker) reading to line follow, it assumes that the processed number represents black. This rules out the need for the programmer to take and input light readings and removes "human error" from taking readings.

