Programming Explorations
Using sensors

Using sensors

You have already successfully written some programs that work in the LEGO Mindstorms RCX brick. So, what comes next? Now you can begin to think about what other sorts of models you could build and the sort of ROBOLAB programs you might write to animate them.

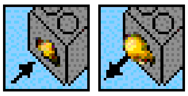
One thing to think about is that LEGO models can have *sensors* attached to them. Sensors can be used to take input from the world. ROBOLAB programs can then be written that use sensor input to change how a model behaves.

- A model that moves might have a touch sensor attached to a bumper so that it can tell when it has hit an obstacle and turn around.
- A model with a light sensor might use it either to move towards a source of light or to move away from it, in the way that some animals behave.
- A model that has a temperature sensor might turn a fan on when the temperature gets too hot, and turn the fan off when it gets too cold.

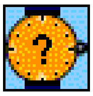
Let's explore how touch sensors work.

The Touch Sensor

The LEGO touch sensors look like the picture on the right. Sensors can be attached to any of the positions marked '1', '2' or '3' on the RCX brick. These are called 'input ports'.

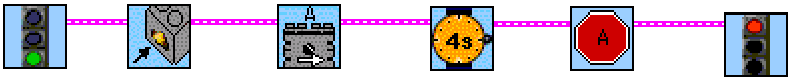


In ROBOLAB, these are the two main symbols used in programs when a model has a touch sensor attached. The first is called 'wait for press' and the second is called 'wait for release'.



You can find these under the 'wait for' symbol in the Functions window.

What do you think that the following program will do if it is used with a windmill model?



Try it out. Take one of the windmill models. Attach the motor to output port 'A' and attach a touch sensor to the position marked '1'. Fix the touch sensor to somewhere on the windmill.

Did the model behave as you expected? Did the motor start as soon as you pressed the Run button? If not, what did you have to do to make the motor start?

Do you understand how the ‘wait for press’ symbol works? If not, ask for some help.

Can you guess how the ‘wait for release’ symbol would work if it were part of a program? Check it out by changing your latest program. Change the program so that the windmill doesn’t run for a fixed length of time but instead keeps running until the touch sensor is released. Here is how you might describe the program in words instead of symbols:

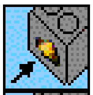
> Start the program.
> Wait until the touch sensor is pressed.
> Start motor A.
> Wait until the touch sensor is released.
> Stop motor A.
> Stop the program.

Run your program several times to check that the motor only stops when you release the touch sensor.

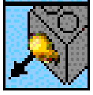
If the touch sensor were to get stuck in the pressed position, do you think the windmill would run forever?

Review

- Programs often need to monitor their environment, and react to it.
- Touch sensors are one of the ways in which a LEGO model might receive input from its environment.
- A program waits at this symbol until the touch sensor is pressed.



- A program waits at this symbol until the touch sensor is released.

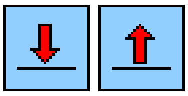


Doing things over and over

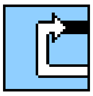
All of the programs you have written so far have run the symbols between the green light and the red light just once. For instance, they made the motor turn for a few seconds, and then stopped.

What if you want to run the same program over and over again, without having to keep pressing the Run button? You might want the motor to run for a few seconds every time the touch sensor is pressed and have this go on all day, without having to be there to re-run the program yourself.

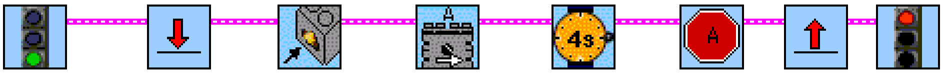
In order to do this, you need to use *a loop*. Here are the two symbols we use for loops in ROBOLAB. They are called a ‘landing’ and a ‘jump’. We always use two of the same colour, such as red.



You can find these pairs of symbols under the following symbol in the Functions window.

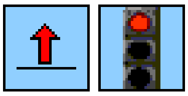


Here is a program that uses a landing and a jump to create a loop in a ROBOLAB program.

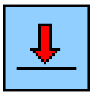


See if you can work out what it will do with a model that has a motor connected to position ‘A’ and a touch sensor connected to position ‘1’. Try it out on a windmill.

One thing that may be a little bit confusing is that the program contains a line between the jump and the stop symbol, but when the jump is reached, we actually go back to the matching landing symbol, so that the program is started again.



The other unusual thing is that the first time the landing symbol is reached, nothing in particular happens – we just move straight on to the next symbol



Summary

Programs often need to monitor their environment, and react to it. Touch sensors are one of the ways in which a LEGO model might receive input from the outside world.

Loops allow us to write programs that perform whole tasks or sub tasks over and over again.