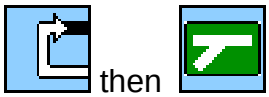
Programming Explorations
Making decisions in a program

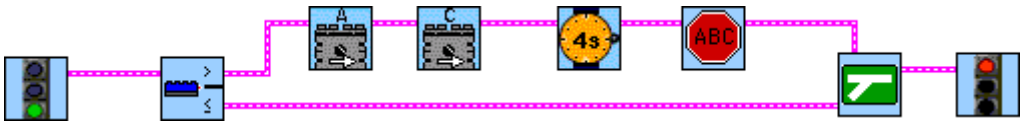
In most programming tasks decisions have to be made. Programs involving control usually make decisions based upon the values received from sensor inputs.

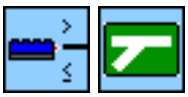
Decisions in ROBOLAB are called ‘forks’ and they can be found by following these icons in the Functions palette:



Fork and merge

Sometimes we need to choose between two possible courses of action. For instance, you might want a vehicle to move out of sunlight into the shade. In order to do this a program would take a reading from a light sensor and if the value is high, turn its motors on for a few seconds. Here is a program to do this that uses two new symbols.





The first new symbol is a light-sensor fork and the second is a merge. A light-sensor fork compares the current reading of the light sensor against a value. The default value is 55 but it is possible to specify any value in the range 0 to 100 by using a modifier.

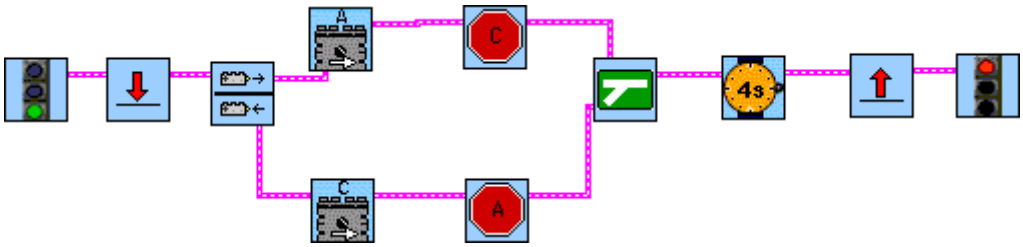
When run, the program will take only one of the two possible paths following the fork. If the light reading is greater than 55 then the top path is followed. In this case that will mean that the motors will be turned on for four seconds. If the reading is less than or equal to 55 then the lower path will be taken. This example includes no actions along that path. Both paths are then rejoined to a single path at the merge symbol.

Try this program with a vehicle. You can add a modifier to the fork if you wish to specify a different light level for the decision. Run the program several times with the vehicle in parts of the room with different light levels.

Can you add a loop to the program so that the light sensor fork is executed over and over so that you don’t have to keep re-running the program?

Touch-sensor fork

The program below uses a different sort of fork: a touch-sensor fork.



It takes one of two different paths depending upon whether the touch sensor is pressed or not. If it is not pressed (the top action) then motor A is turned on. If it is pressed, then motor C is turned on. Both paths then merge and the program pauses for four seconds before the test and actions are repeated. The program loops forever.

Try it out with a model.

More modifiers

Further modifiers can be used to indicate various things:

These indicate which input port a sensor is attached to. They are useful if you are using more than one sensor on a model
These indicate what power level a motor should have. These allow you to control the speed of a model.

Project challenge

Design and build a LEGO model and then program it to complete one of the following projects, or come up with something similar of your own.

- A line follower. The model must follow a black line on a white background. It must be capable of following a line with both left and right turns.
- A wall follower. The model must move along a wall, staying as close as possible to the wall. The model must be able to follow turns in the wall. It only needs to follow in one direction.
- A light follower. The model must find a bright source of light in the room and move towards it. You need to bear in mind that the source might move.

Spend time in discussing both the design of a model and the program strategy you will use to solve the task. It is important to appreciate that both of these need to be considered together.