RCX Programming Tutorial

Big Blocks

Big Blocks are sets of commands that are predefined for different robot types. You should not use them in your program. We want you to learn basic programming and therefore, we would like you to create your own blocks from more basic commands.
However, you are encouraged to look at the Big Blocks to learn how to make your own blocks. For example, one of the blocks in the Roverbot secion is Turn Left shown below.

Click on the upper left corner of any big block (the gray square) to expand it and see small blocks inside.

Small Blocks

Here are the basic commands that you will be using to program your robot.

Set Power

This is the first command that you should use in your program. This command sets the power level for the motors (1 is usually too low, 8 is the maximum). Remember that the battery power affects the speed of the motors (i.e. 8 can be really slow if your batteries are weak).

Set Direction

This is the second command that you will need. It sets the direction that the motors turn. You should download a sample

program to determine which direction is forward and which is backwards, as it depends on the orientation of the wire connection.

On

Turns motors (connected to ports A, B, or C) on (i.e. moves robot forward or backwards). The motors are on until you turn them off using the **Off** command below.

Off

Turns the motors off. Note that there is a difference between the coast and break option. Braking means that in addition to stopping the motor, brakes are applied to prevent the robot from moving. Coasting stops the motor but allows the robot to still move until it stops by itself or runs into something.

On For

Allows the robot to move for a specific time period (in seconds).

Beep

Causes the robot to beep a specific number of times. This is a good debugging tool.

Tone (music)

Similar to beep, but plays a different tone for a specific period of time. Can be used to write music!

Send IR Message

This command allows the RCX to send an infrared message to another RCX.

Clear IR Message

This command clears out any previously received message from another RCX. Note that you have to clear out the current message in order to be able to receive a new message. For example, in your wheelchair program, you have to stop after receiving signal “1” from the master controller. You should start moving again after receiving signal “2”. So, in order to be able to receive message “2”, you should clear message “1” first.

Counter Variables

These commands can be used to keep track of the number of times certain events occur (for example how many times the touch sensor is pressed or the robot has turned). At the beginning of the program, the counter is always 0. To use, simply place the "Add to Counter" button on top of the command(s) you want to count. It is a good idea to reset the counter when you are done using it (so if later in the program, the robot needs to count something else, it will start from 0).

Wait (does not mean "stop & wait," only wait)

Wait For

Does not mean stop! Instead causes the robot to wait a specific period of time before executing the next command.

Wait Until

This command allows the robot to "Wait Until" the light sensor reaches a specific value before executing a command.

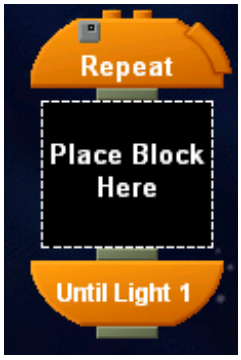
Repetition Commands

Repetition commands allow you to create more advanced programs.

Repeat (# of times)

To repeat a set of commands, simply place them in between the "Repeat" and " the number of repetitions " blocks.

Repeat Until



Execute a series of commands as long the light sensor reading is outside a certain range. Once the light enters the predefined range, the program exits the loop. This is just an example of a repeat command. You can use a touch sensor or a variable counter as well here.

Repeat While

Execute a series of commands as long the variable counter is within a certain range. Once the counter exceeds the predefined value, the program exits the loop. Again, you can instead use a light or touch sensor here.

My Blocks

"My Commands" allow you to create subroutines for your programs. Subroutines are essentially small parts of programs. The benefits of using subroutines include the following:

- programs are easy to read,
- time is saved since you only need to create the code once to use over and over
- multiple people can work on the same project (industry).

Sensors

For all sensor watches, you can create two lists of program commands, depending on the state of the sensor. Note - you do not have to write code for both conditions. If one is left blank, then the program will revert to the main program. The sensor watches have to be placed at the top of a program thread.

Light Sensor

Use the view button on your RCX to determine the light intensity in the location your robot will be running. Use these values to set the range for dark and bright. This process is called calibration and you should do it often since light conditions change all the time.

Touch Sensor

Select the port (1-3) that the touch sensor is attached. Usually it is easiest in a complicated program to only write code for the "Press" condition.

Infrared Sensor

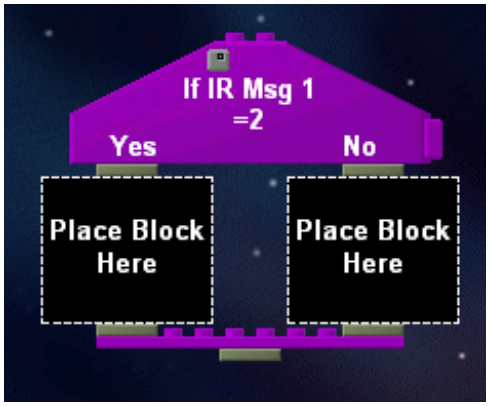
This command allows you to program your robot to do certain commands when it receives an infrared signal.

Counter

The counter can be used to program the robot to do certain tasks after or before a specific set of events occurs. For

example, you can count the number of times your robot hits the wall and stop after you reach 10 collisions.

Yes or No Command



This is a conditional command. If the robot receives an IR message "2", the left set of commands are executed. If the robot receives an IR message of "1", the right set of commands are executed.
In addition to the IR message, the "Yes or No" command can be conditioned on the state of the touch and light sensors.
Note that compared to the sensor watches shown above, the "Yes or No" command does not have to be placed at the top of a program thread. Instead, if can be placed in the middle of any thread.