

A Short Tutorial on Using Robolab Software with the LEGO RCX brick

Tutorial Part One : Opening RoboLab

Welcome to Robots and the RoboLab software. RoboLab is the application you will use to program your robots.

To open RoboLab, click the ROBOLAB 2.5 shortcut on your desktop. A window will open giving the options of Programmer or Investigator as illustrated in **Figure 1**. Select Programmer.

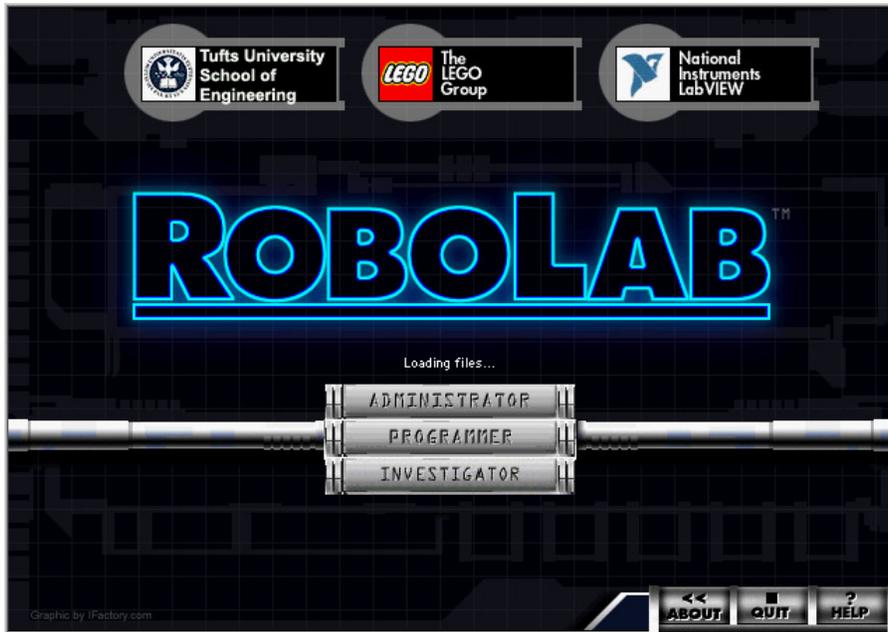


Figure 1: First RoboLab screen.

Upon selecting Programmer, a new window appears showing Pilot and Inventor levels. For our program, we will be using the Inventor 4. Select the Level to be Inventor Vault, Theme as My Programs. Your window should look like **Figure 2**.

Short Tutorial on Robolab



Figure 2: Selecting a program

Upon double clicking on the LineTracer.vi file, the RoboLab workspace for programming. We will explain all of the features of the workspace later.

Turn the robot on by pushing the red On-Off button. Select Program 5 to download to by pushing the Prgm button until the display shows a 5. Place the robot in front of the IR

Transmitter. Click the  to download the program to the robot. A progress bar will appear and when the download is successful, the robot will play a little song.

Congratulations. You have just downloaded your first program to your RCX. The next page discusses the RoboLab workspace.

Tutorial Part 2

Robalo Workspace

Now that you know how to download a program to your RCX, you're ready to learn how to program your robot

Open RoboLab and open up a workspace by double clicking on Inventor 4 in the level section.

Short Tutorial on Robolab

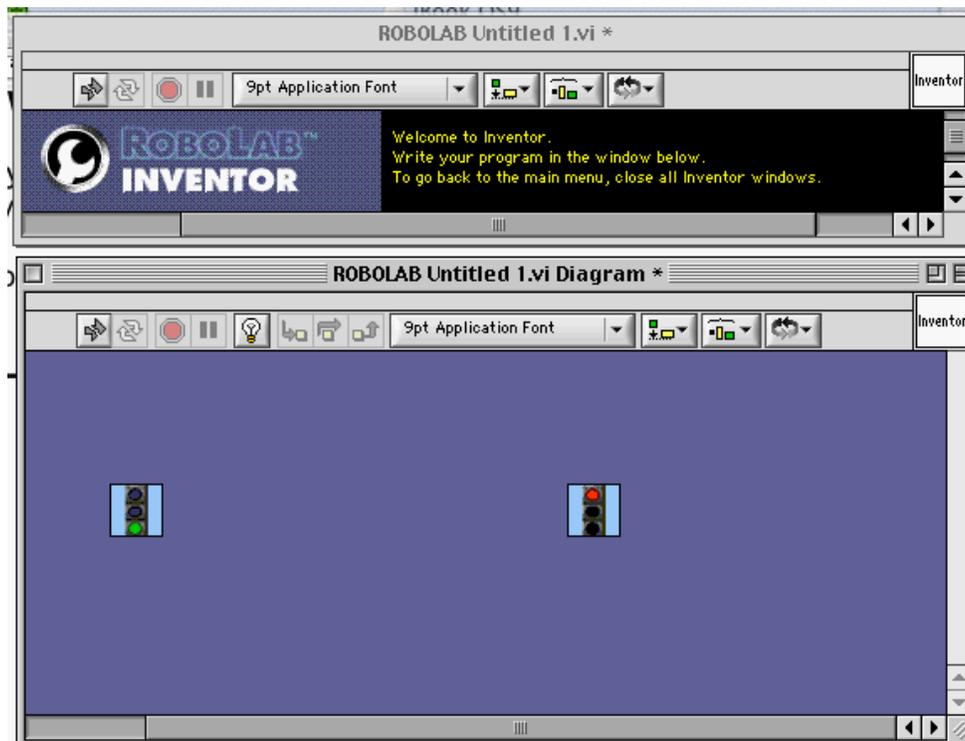
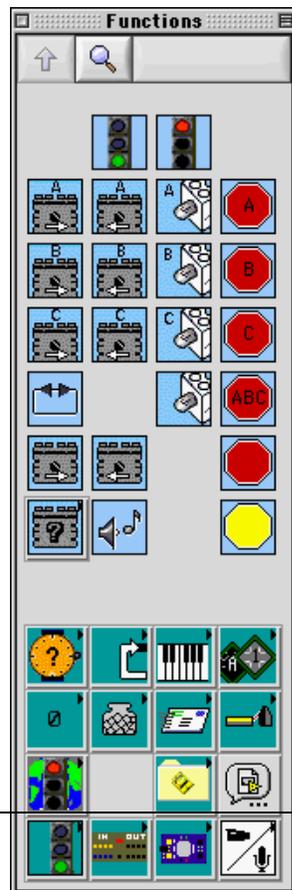


Figure 1: Beginning View of Workspace

All programs start with a green stop light and end with a red stop light. To move the red and green stoplights to where you want them, press the SPACE BAR until a purple arrow outlined in black and white appears. Now you can click and drag the stoplights to where you want them.

Now try adding a function to your program. The function menu is shown on the next page:



If you can't see your function menu, in the workspace, click on Window and then Show Functions Palette. Try adding the function to turn on the motors.

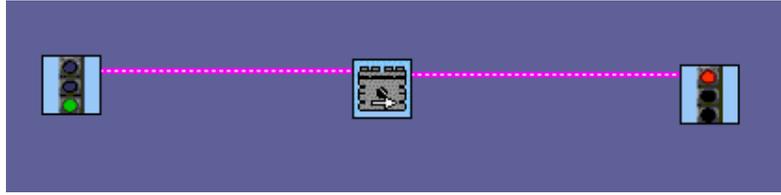


Right click once on the  and then drag it onto your program somewhere between the green and red lights. To place it, drag it to where you want it to go, and then right click once again.

Now you need to connect the functions together. To do this, press the SPACE BAR until an icon comes up that looks like a spool of thread.  The purple arrow and spool are also available by clicking on Window and then Show Tools Palette in the workspace. Once your mouse becomes the spool icon, move the mouse over the upper right corner of the green stoplight. The corner will start to blink and it will say "End" out to the side. Right click once to start the thread and move the mouse over to the upper left corner of the motor icon. The corner will start to blink and say Begin. Right click once again to connect the thread to the motor. Try this again connecting the upper right corner of the motor to the upper left

Short Tutorial on Robolab

corner of the red stoplight. If the thread was connected correctly, it will turn pink. (See Figure below).



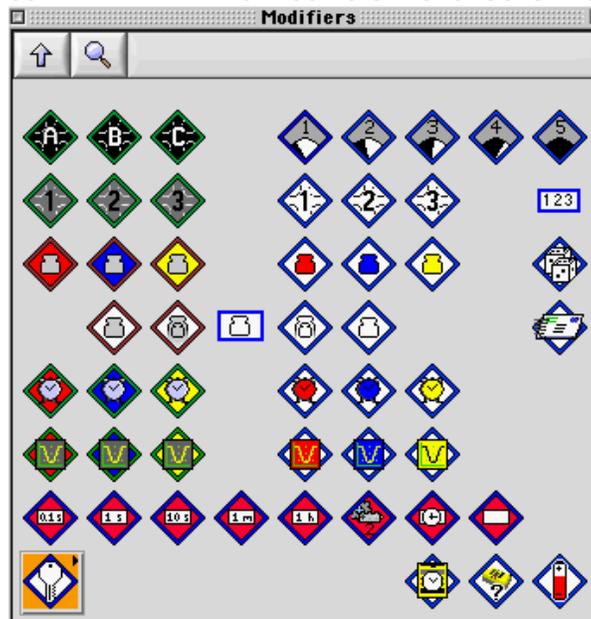
Connecting a motor function

If you look at your RCX, you will see that there are three ports labelled A, B, and C. These are the ports used by the motors. By default, the motor function turns on all 3 ports at full power. Double click on the motor icon in your program to bring up information about it. (You can do this with all icons). We are going to only turn on ports A and C by adding attributes to the motor icon.

To do this:



Click on the modifiers icon . The modifiers menu looks like this:



Now right click on the A port and place it on the workspace close to the motor icon.

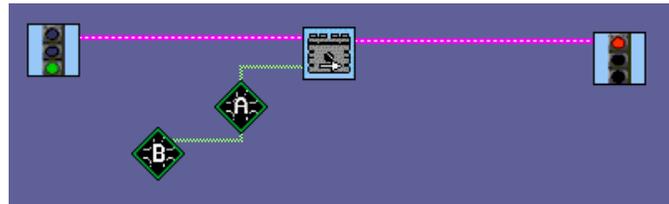
Do the same for the C port.

The program may have placed connecting wires in for you. If they were placed correctly, they will be a light green color. If not, then they will be black and white slashed lines. To correctly place the wires, delete any that it already created. To do this, press the SPACE BAR until the purple arrow comes up, then right click one of the wires, and press the DELETE key. Repeat this until all the broken wires are deleted. Another (easier) way to do this, is to click on Edit and then Remove Broken Wires

Short Tutorial on Robolab

Now to properly connect the attributes, press the SPACE BAR until you get the spool and mouse over the lower right corner of the motor icon until it blinks and says Ports. Right click once and move the mouse over the top of one of the port attributes (it will blink and say Ports). Right click to connect them.

Now, click on the bottom of the port attribute you just connected (it will blink and say Additional Ports, and mouse over the top of the other port attribute (it will blink and say Ports). Right click again to connect them. Both connections should be green. Your program should now look something like this:



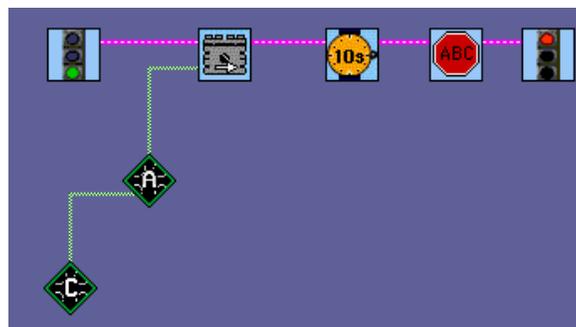
All modifiers are in this Modifiers menu if you would like to add others (i.e. changing the power level, etc.)

Now try adding a Wait function to your program. To get from the Modifiers menu back to the Functions menu, click on the Arrow Up button. The Wait For function looks like an



orange stopwatch with a ? in it. Right click on it to go to the Wait For Menu. Now choose one of the stopwatches with numbers in them (i.e. Try 10s). Right click on it and place it on your workspace. You will have to delete/reconnect some wires in order to connect the Wait For function to your program. Connect it after your motor Function but before your red stop light function. One rule you must remember, if you start something you must stop it. Add a stop sign for all motors. Just placing the function on top of existing wires does not automatically connect them; you have to re-arrange the wires.

Your program should now look something like this:



Now that you know how to add and modify functions, try adding more to your program (such as stop the motors or switch directions, etc).

To run your program, click the white arrow  to download to your RCX and then on the RCX press the Run button.

Congratulations. You have just programmed your Robot!

Tutorial Part 3

Code Examples

Now that you know how to program and download a simple program to your RCX, let's learn some of the basic programming structures. This section of the tutorial will highlight a few of the major ones. For more information, please refer to the reference pages.

Turning Motors On and Off

You can turn a motor on two ways.



The first is to use the command with the name of the motor on it like

The second way is to turn it on by using the general Motor Forward and using a Modifier to



specify the motor. This looks like:



To turn a motor off, simply use the command, specifying which motors to turn off.

Splitting Tasks

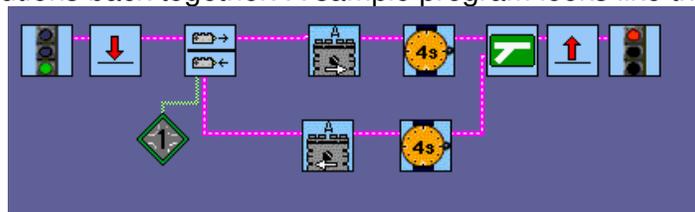


If you want your robot to do two things at once, you can use a task split. This will allow the robot to execute both pieces of code that come from it.

Conditionals

If you have a conditional, such as the use of a touch or light sensor, your code will branch

in two directions, based upon the value of the sensor. The  command will bring the two separate executions back together. A sample program looks like the following:



Here the motor A goes forwards and every 4 seconds the RCX checks to see if the button has been pushed. If it has, the motor reverses for 4 seconds.

Wait For Command

Description: The amount of time that a motor is on.

To do this: Click on the watch and choose from the time selections which appear in the menu bar. The choices are 1,2,4,6,8 or 10 seconds. Your watch window will look like figure 1.

Short Tutorial on Robolab



You can also select the  command which allows you to specify an amount of time. The default is 1.

Another feature of the Wait For commands is to specify a random time. You can connect a numeric constant  which will specify the interval of the random number.

Looping



Looping will run a piece of code more than once. You can use the  commands to run a piece of code. To specify the number of times it will execute, add a numeric constant from the loop. The default is 1.



Making a Child's Toy

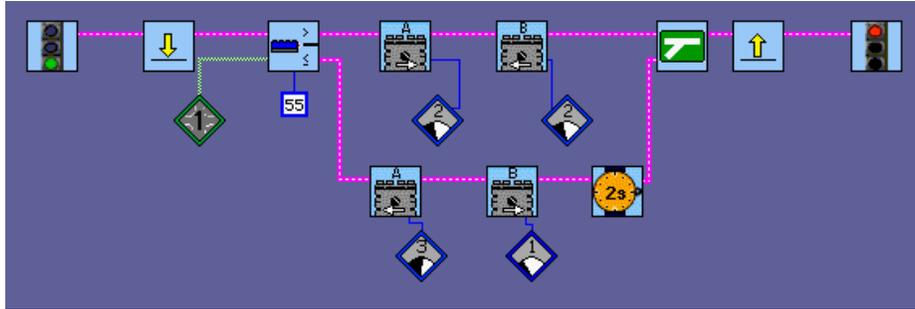
A little car that drives along a white surface slowly until it 'sees' the edge. It stops and reverses back turning as it does so.

Steps:

1. Open RoboLab and select "Investigator". Open a new project. Select level 5.
2. Between the start and stop signs we need to create a branch. If the car has NOT 'seen the edge' it goes forwards with both motors moving slowly at the same speed.
3. The programme then jumps back to the beginning again using a 'jump' and a 'land'

Short Tutorial on Robolab

4. If the light sensor 'sees' black the programme takes a branch making the motors reverse for 2 seconds.
5. To make the car turn one motor operates faster than the other one.
6. After the programme waits for 2 seconds the branch rejoins the main programme and then jumps back again to check on the light sensor.
7. The hard part of this programme is telling the RCX what to look for. On the RCX press the view button and select input 1. Slowly run the car over the white/black part of the paper and see how the numbers on the screen change. You must enter a number where this programme has '55' so that it is smaller than the white reading and higher than the black.



Think your way through this programme before you enter it into your computer and try it. Feel free to modify the code.

Competition

How would you modify this programme to have the car run along the ground and turn if it hits something?

You have a touch sensor, so change the programme, modify the car, and try it. If you could use two touch sensors ask for them!!

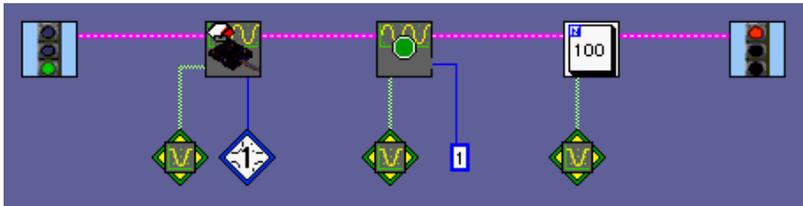
Data Logging

For our first exercise in scientific investigation we can log data. This means the RCX will collect information from the world around it and store that information. We are fortunate enough to have some very expensive probes that connect to the RCX to collect data.

This example tells the RCX

1. To begin
2. That it will be collecting data from a sound sensor [there are alternates if you do not have a sound sensor] connected to port 1 and that the data is to be stored as 'yellow data' This is just a way to separate it from other probes that could be 'red data' or blue data'
3. The RCX is told to collect the data each second and keep it as 'yellow data'
4. After 100 data points have been collected [100 seconds] the programme stops.

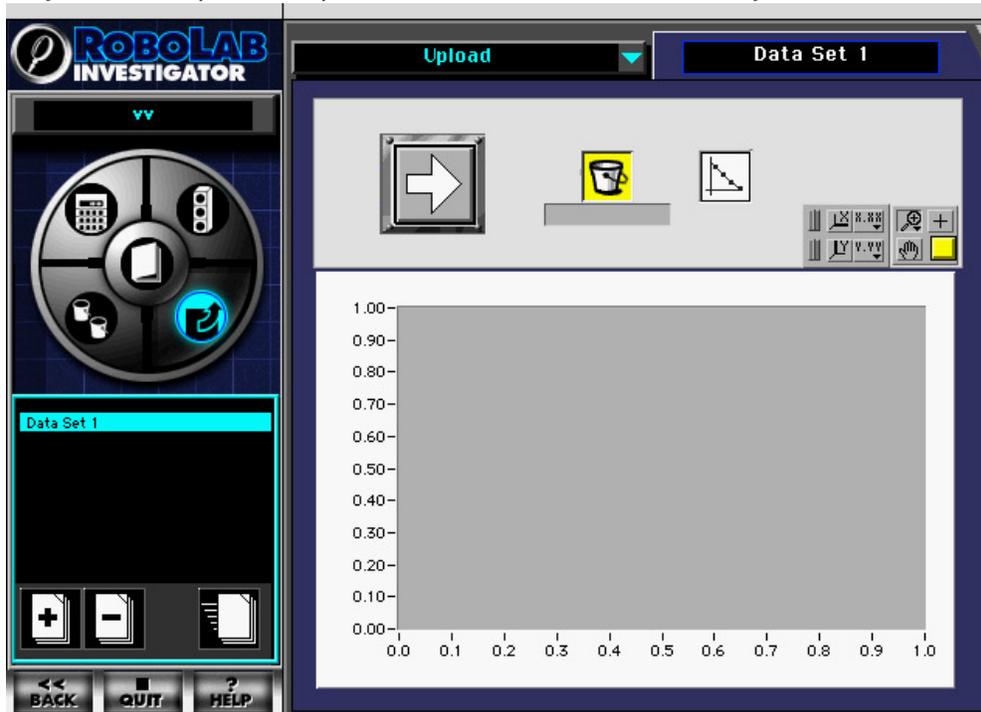
Short Tutorial on Robolab



Next we de-select 'full size' from the 'window' panel and then we can see the command centre at right. If we select the upload arrow and then the big white arrow the data will be sent from the RCX to the tower connected to the computer. We choose the 'yellow' bucket and the data is displayed.



If we select then small graph button we can see the actual figures and they can be copied and pasted into Excel for further analysis.



We hope your reports would be written in Word with the data from the RCX analysed in Excel.

Although the programming is not very difficult, try setting the RCX for another probe and swap with another group.

You can collect only 1200 data points so you need to think carefully about how often you will collect data.

To run an experiment over a whole week [7 days x 24 hours x 60mins x 60 secs = 604800secs] you could only collect data every [604800 / 1200 = 504sec] 8 minutes.

You also need to think, "Do I want 1200 data points?" "Would one every 15 minutes be enough?"

Your Experiment

Design a programme that would collect the temperature and light intensity in the classroom over a 24-hour period. The RCX would need 2 sensors of course.