Lego Robot Tutorials
Programming and Motors

Program Your Robot to Move

In this activity you will write programs that make your robot move. Along the way you'll start learning RoboLab, the software we use to program robots. No previous experience is required! Go to this page on the Web site to see the robot drive around the penguin, pursued by Razzle the ferret.

Before starting this activity, you should *Build a Robot* and *Get Started*.

Autonomous vs. Remote Controlled Robots

Robots can be divided into two categories depending on how they are controlled.

Autonomous robots are controlled by a computer program. The program reads data from the robot sensors, makes decisions on what to do, and sends commands to the robot motors. A human programs the robot before it starts running.

Remote controlled robots are controlled by a human operator. The operator uses his or her own eyes and ears, makes decisions, then sends commands to the robot motors to control its motion.

The Lego Mindstorms™ robots that we use are autonomous robots: you must program them to make them go. Factory robots that assemble cars, cruise missiles, robot vacuum cleaners, robot lawn mowers, and some robot toys are other examples of autonomous robots. Can you think of any others?

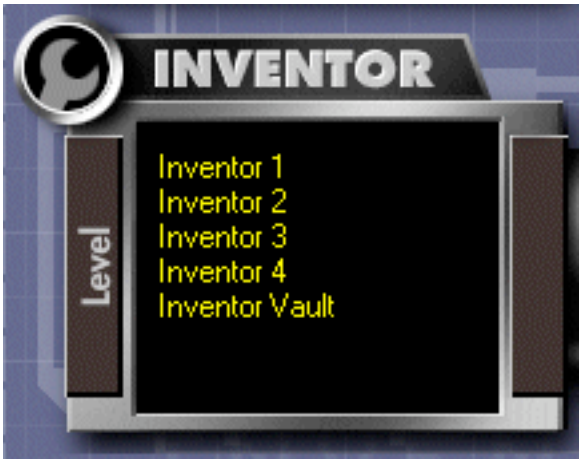
Battle Bots on TV are a famous example of remote controlled robots. Other examples are the robot submarine in the movie Titanic (a real world robot), robot arms and manipulators for dangerous chemicals, and underwater robots that recover the flight recorders (black boxes) of planes lost at sea.

Start RoboLab, Open the Programming Windows

1. Start RoboLab by double clicking the RoboLab icon  on your desktop. If you can't find the RoboLab icon, ask your coach where to find RoboLab on your computer.

1. When the RoboLab main menu appears, click Programmer.

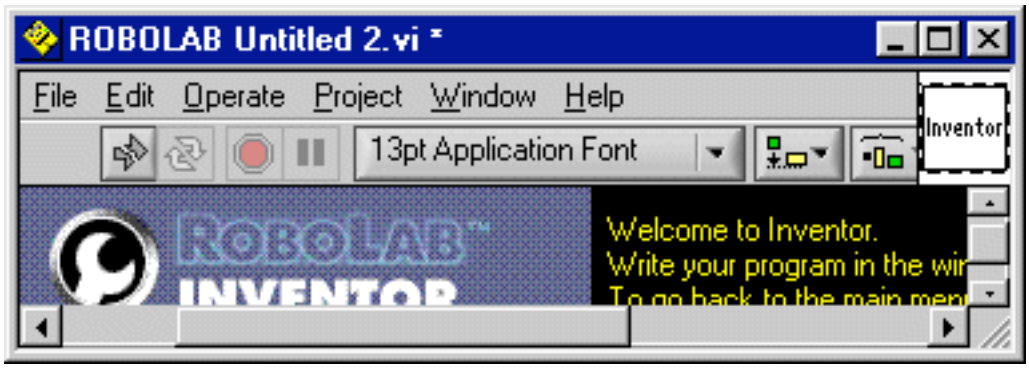
1. Double click Inventor 4 to open the programming windows.



NOTE: This screen provides a choice of various "levels". The lower Pilot and Inventor levels have fewer features and are meant to be easier to learn. But some of the functions we need are only found on Inventor 4. Most students don't seem to mind having more things on the menus, so we go straight to Inventor 4.

The Panel Window

When you double click **Inventor 4**, three windows open. The top left window is called the Panel Window. This doesn't actually do anything, but if you close it the other windows close too. Click the minimize button to get it out of the way if you like.



Panel Window

The Diagram Window

The lower left window is the diagram window. This is where you create your computer program. A RoboLab program is a diagram of connected functions or icons that tell the robot what to do. This type of programming is called graphical programming or iconic programming.

RoboLab puts two functions in the diagram for you automatically: Begin  and End  . These are always the first and last functions in your diagram.

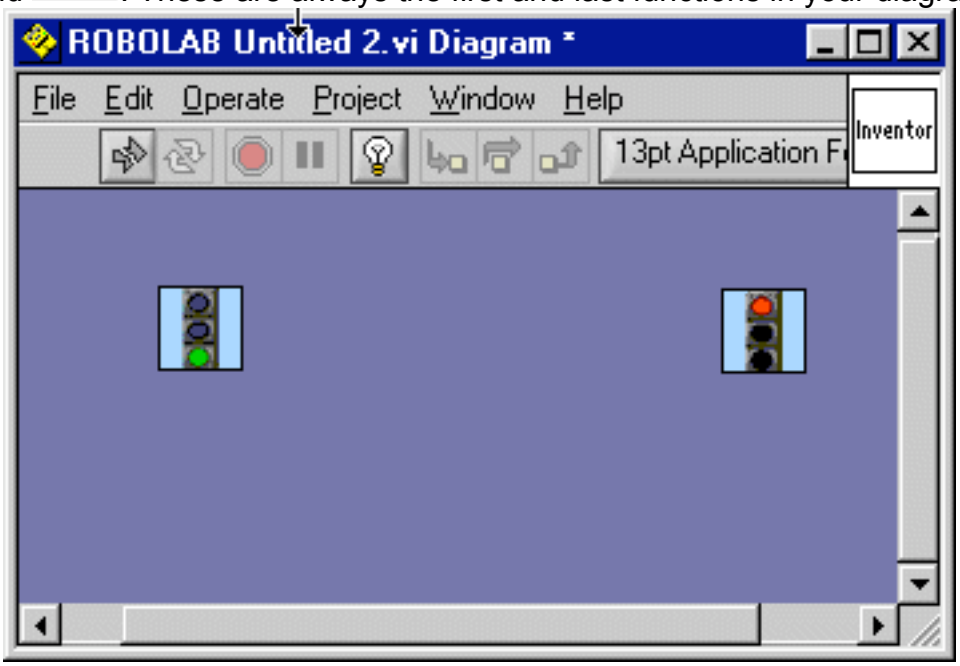
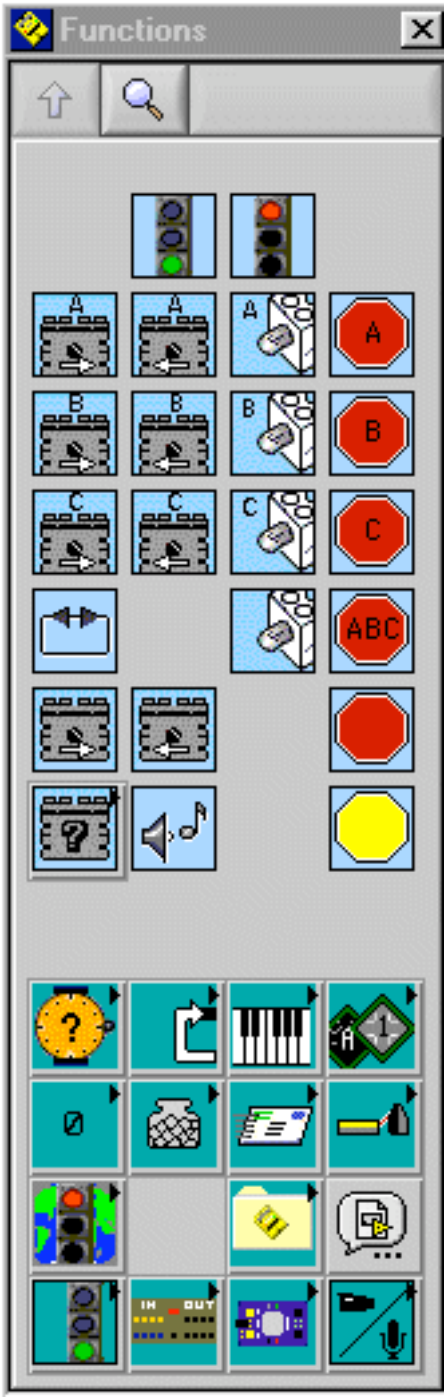


Diagram Window

The Functions Palette

The right window is the Functions Palette. An artist's palette is a board with many colors of paint to choose from. The Functions Palette contains all the functions you can choose from to create your robot program. If the Functions Palette is not visible, click the Diagram Window. If it still doesn't appear, choose **Window > Show Functions Palett**e from the Diagram Window.

In these activities we'll show screen shots from RoboLab 2.5. The Function Palette looks a bit different in earlier versions of RoboLab, but you should still be able to find the functions you need. We won't use any function not found in all versions of RoboLab.



The Functions Palette

The Tools Palette

The last programming window we need is the Tools Palette. It doesn't open automatically, so open it with Window > Show Tools Palette from the Diagram Window. Use the Tools Palette to select the cursor tool you need. There are different tools for different jobs, as we'll see shortly.

Connect the Motors to the RCX

The yellow RCX brick has three motor *outputs*: A, B, and C on top. Each output can control one Lego motor. For this activity, check that the A port is connected to the motor for the left wheels with a short wire, and the C port is connected to the motor for the right wheels.

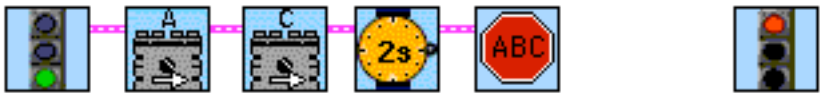
The way the wire is plugged into the output port and the motor affects which way the motor turns. In certain combinations, the motor will go in reverse when you tell it to go forward! To avoid this problem, make sure the wire comes out toward the back of the robot from both the motor and the motor port.

Write a Program

Now you are ready to write a program! This program will make your robot go forward for 2 seconds, then stop.

Select the arrow tool  from the Tools Palette. The arrow tool is used to drag and position icons.

Drag icons from the Functions Palette into the Diagram Window to make the diagram show below:



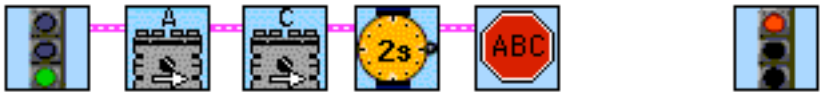
You can read this diagram as:

- Begin
- Start Motor A
- Start Motor C
- Wait for 2 seconds
- Stop All Motors
- End

To find the **Wait for 2 Sec**  function, click the **Wait For**  button on the Functions Palette. This brings up the Wait For Palette, which contains various wait functions. Click  to return to the main palette.

TIP: When you drag an icon from the Functions Palette next to an icon already in the diagram, RoboLab 2.5 will automatically connect the two icons with a pink wire if they are close enough. Earlier verisions of RoboLab don't autowire. Don't worry if some of the wires aren't there.

The Meaning of "Wait"

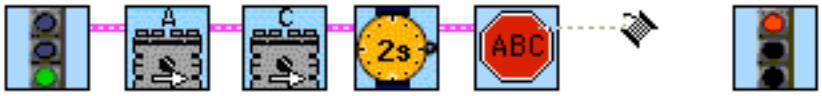


Wait for 2 Sec  tells the program to wait for 2 seconds before going to the next step, which is **Stop All Motors**. It's the program that waits, not the robot. During this time the robot is still going. So combined with the start motor functions, this really means "go forward for 2 seconds". If you take out the **Wait for 2 Sec** function, the robot will stop immediately.

Connect the Icons

Choose the Connect Wire tool  from the Tools Palette. The cursor changes to a spool of thread, which you use to connect the icons with wires. When the program runs, it moves from one icon to the next following the path of the wires.

Pick two icons that aren't connected. Click and hold the upper right corner of the first icon, and drag to the upper left corner of the second icon. The picture below shows the last wire being added. Wire each icon to the next one.



NOTE: You must connect each icon to the next. One long wire through several icons might look exactly the same in the diagram, but would only connect the first and last icons.

TIP: For a shortcut, click the space bar to switch between the Connect Wire tool and the Arrow tool.

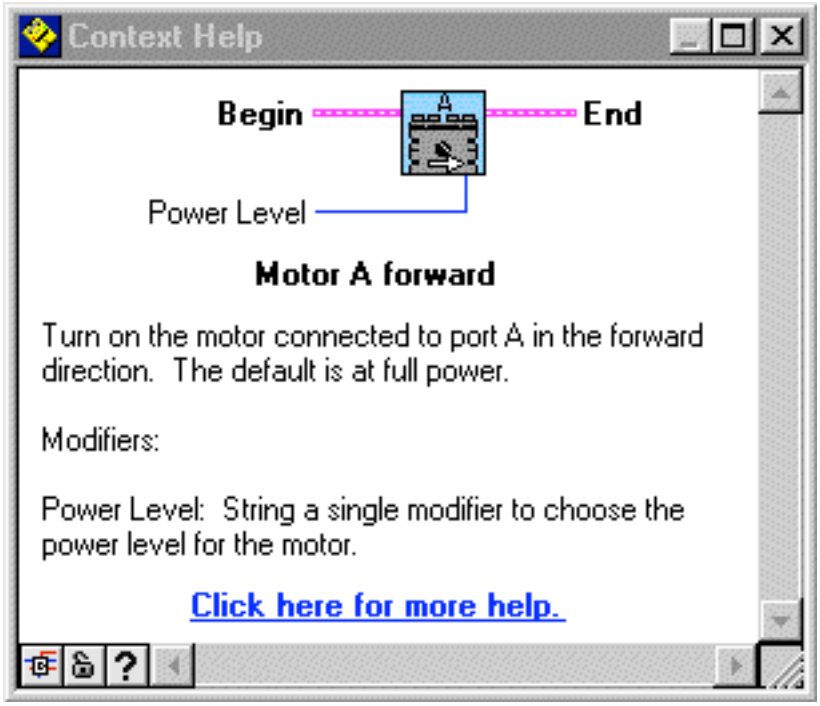
Fix Broken Wires

Wiring takes a little practice. If a wire is not connected properly it is called a broken wire and shown in black and white diagonals  . Use the space bar or Tool palette to turn the cursor into an arrow  . Select the broken wire by clicking it, then remove it with the Delete key. Or select **Edit --> Remove Broken Wires** from the menu bar of the Diagram window to remove all broken wires.

If there are missing or broken wires in your diagram, there will be a broken Run arrow  near the upper left corner of the Diagram Window. Use **Edit --> Remove Broken Wires**, then add any needed wires with the Connect Wire tool  .

Context Help

RoboLab has an excellent online help system. Open the **Context Help** window by choosing **Help > Show Context Help** from the Diagram window. Now point the cursor at the **Motor A Forward** icon in your program. You will see something like this:



This window explains everything you need to know about the **Motor A Forward** function. The **Begin** wire shows where to connect the wire from the previous function. The **End** wire shows where to connect the wire to the next function. **Power Leve**l shows where to connect a modifier, in this case a number to set the power level. More on modifiers in future activities. **Click here for more help** shows sample diagrams that use this function.

You can even get context help by pointing to functions on the Functions Palette. Use Context Help to explore RoboLab.

Save Your Program

Now is a good time to save your program to disk. Experienced programmers save their programs often - every few minutes when they are making changes. That way they don't lose everything if the computer crashes or RoboLab locks up. Inexperienced programmers don't always save often enough, until they get experience losing their work!

Use **File > Save** to save your program. Give your program a descriptive name so you know what it is next time, like **foward2sec**. You can save your program to any of the places below. Ask your coach which is best for your site.

file server - A file server is a central computer that stores files for users anywhere on a network. If your site has a file server and you have a personal directory there, this is the best place to store your program files, because they are safe and you can access your files from any computer on the network. You may need to log in with a username and password to access the file server. From **File > Save**, choose the file server disk in Save In:, choose your personal directory, name your file, and click Save.

floppy disk - If your computer has a floppy disk drive, you can store your programs on a floppy disk. That way you can take the floppy disk to whatever computer you are using each day, and your programs will be safe between club meetings. Write your name on the floppy label, and have your coach keep your floppy in a safe place between club meetings. Make a second floppy once in a while for backup, because sometimes floppies go bad. From File > Save, choose the floppy disk **in Save In**: (for example "3 1/2 Floppy a:"), name your file, and click **Save**.

local hard disk - You can also save to your computer's local hard drive. This is fine for your own personal computer, but has some drawbacks for a club. First, you have to use the same computer each club meeting, because you can't read your files from any other computer. Second, there is a chance that another user will accidentally delete your files between club meetings. Ask your coach where is the best place to store your program files on the local hard disk. Many sites have a special directory for student files. From **File > Save**, go to the student directory and create a new folder just for your files. Name your personal folder with your name or initials. Select your personal folder, name your file, and click Save.

For the three options above, use **File > Open** in the Diagram window to open your program file the next time you need it. Navigate to the correct directory.

RoboLab Program Vault - RoboLab has a special place on the local hard disk you can use for program files. This is the default location for saving files with File > Save. You can save your programs there, but other club members who use this computer may save files in the same directory, which will be confusing. You can

open files from this directory from the Pilot and Inventor screen you pass through when starting RoboLab. Select **Inventor Vault** and **My Programs**.

Download Your Program

When a diagram has no missing or broken wires, there will be a solid Run arrow  near the upper left corner of the Diagram Window. Correct any wiring problems until you see a solid arrow.

You can store up to 5 different programs in your robot at once. This can be useful in competitions. You select the program slot to use with the **Pgrm** button on the RCX. By default slots 1 and 2 are locked.

Place the RCX in front of the IR tower, turn on the RCX, and select program 3, 4, or 5 with the **Pgrm** button. This determines which of the five program slots your program will download into. By default slots 1 and 2 are locked.

Click the Run arrow  . The program begins downloading. If you get a message saying the firmware will now download, wait 4 minutes for this to happen. This should only happen if the batteries have just been removed or are loose. If you get an RCX communications error, see the page Troubleshooting RCX Communications.

TIP: The unlock program slots 1 and 2 so you can use them, turn on your robot and place it in front of the IR tower. Go to the RoboLab main menu (start up screen) and click **Administrator**, then click the **RCX Settings** tab at the bottom of the screen. For **RCX Programs 1 & 2**, select "Unlocked".

Run Your Robot

Put your robot on a flat surface with plenty of room in front of it. Press the Run button. The program in the currently selected program slot will run.

Does your robot go forward 2 seconds and stop? If not, check the following:

- If your robot spins or goes backwards, check that the motor wires face toward the back of the robot where they come out of both the motors and the RCX. It also works if all wires come out toward the front, but not the sides.
- Check that the motor for the left wheels is connected to port A, and the motor for the right wheels is connected to port C.

Troubleshooting RCX Communications

Sometimes you get an RCX communication error when trying to Test RCX Communications or download a program. Here are some things to try:

1. Make sure the RCX is on.
2. If the RCX won't turn on, open it up by pulling apart the yellow and gray halves. Make sure the batteries are in the RCX, correctly oriented, and charged. Turn it over. If the batteries fall out, they are too loose. Carefully bend out the battery contacts using tweezers.
3. Check the battery in the tower is installed, correctly oriented, and not dead.
4. Verify the cable is plugged into the correct COM port.
5. Verify that the correct COM port is selected in RoboLab.
6. Try a different COM port.
7. On USB systems, unplug the USB cable from the computer, wait a few seconds, then plug it back in again. This resets the USB bus.

If none of these work, start swapping out hardware pieces to identify the source of the problem. Only change one thing at a time so you can isolate the problem.

1. Try a different robot.
2. Try a different IR tower.
3. Try a different serial cable.
4. Try a different computer.

Challenges

1. Forward and Back – Make your robot go forward for 2 seconds, then backward for two seconds, then stop.

TIP: Use **Motor A Reverse**  and **Motor C Reverse**  to make your robot go backward.

2. Spinning – Make your robot spin to the left for 4 seconds, then go forward for two seconds, then spin to the right for 1 second, then stop.

3. Around the block – Make your robot go forward for 2 seconds, turn to the right for 1 second, and repeat this four times. Don't make either wheel go backwards.