RoboLab[1] Icon Glossary

The RoboLab user interface employs a drag-and-drop methodology and consists of a series of menus, or *palettes*, from which users select icons, or *commands*. They drag selected icons to a canvas on which they construct a program by stringing a series of icons together using a pink "wire".

This **glossary** contains the icons which are used when first learning the interface.

The top-level **functions palette** is shown below. From this top-level palette, the user can select one of 22 command icons or click on one of 15 menu icons that opens other subsidiary palettes, as indicated by the annotations on the figure. The subsidiary palettes used in the introductory RoboLab exercises are: **Wait For**, **Modifiers**, and **Structures**. For each palette, a table of icons used in the introductory RoboLab exercises is shown on the following pages.

Functions	Wait For	Modifiers	Structures

---
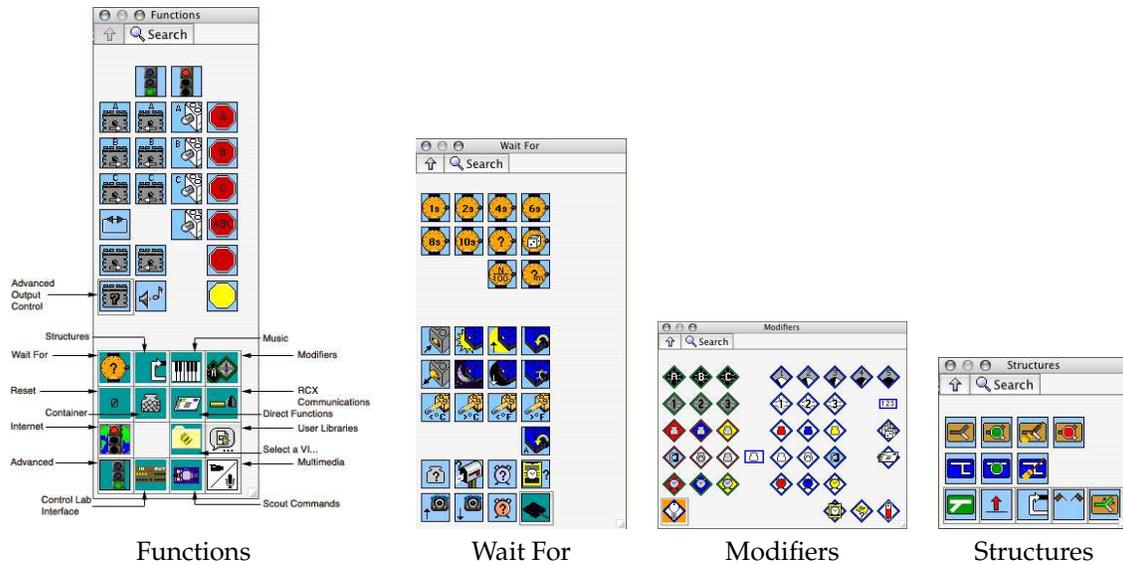
[1] For RoboLab v2.5.4–v2.9

Functions palette

The icons shown below are include on the top-level **Functions** palette. These include basic functions for beginning and ending a program, and for turning on and stopping motors or lamps (whichever is plugged into an output port, labeled A, B or C).

begin program	end program	
motor A forward	motor B forward	motor C forward
motor* forward		
motor A reverse	motor B reverse	motor C reverse
motor* reverse		
stop A	stop B	stop C
stop all outputs	stop output*	
lamp A on	lamp B on	lamp C on
lamp* on		

* indicates that a modifier is necessary to apply the function to one output port (A, B or C); the default is to apply the function to all three ports.

Wait-For palette

The **Wait For** palette contains commands that are all used to indicate that the RCX should wait for a specific condition before continuing to the next command. Each icon specifies a particular condition. Note that this does *not* mean that the RCX is necessarily stopped (not in motion)—it only means that it will not receive new commands until the specified condition is met; in other words, if the robot is going forward prior to receiving a "wait for" command, then it will continue going forward until the "wait for" condition becomes true. The icons shown below are included on the **Wait For** palette.

	wait for 1 second		wait for 2 seconds		wait for 4 seconds
	wait for 6 seconds		wait for 8 seconds		wait for 10 seconds
	wait for N* seconds		wait for (N/100) * seconds		
	wait for light **		wait for dark **		
	wait for push **		wait for release **		

* indicates that a modifier must be included to define the value of N (as in "wait for N seconds" or "wait for N one-hundredths of a second"

** indicates that a modifier is necessary to define which input port (1, 2 or 3); the function is applied to

Modifiers palette

The **Modifiers** palette contains icons that are used to modify other icons. These are typically attached with a blue wire to function, structure or wait-for icons, and do things like indicate an input or output port, or indicate the number of times for a counter-controlled loop to execute. The icons shown below are included on the **Modifiers** palette.

	output A		output B		output C
	input 1		input 2		input 3
	power level 1		power level 2		power level 3
	power level 4		power level 5		numeric constant

Structures palette

The **Structures** palette contains icons that are used to implement program control structure, such as branching, looping and defining subroutines.

The icons shown below are included on the **Structures** palette and are used for working with **Subroutines**.

	create subroutine	
	run subroutine	

The icons on the bottom row of the **Structures** palette are used to access the subsidiary palettes:

Forks palette   **Jumps** palette   **Loops** palette
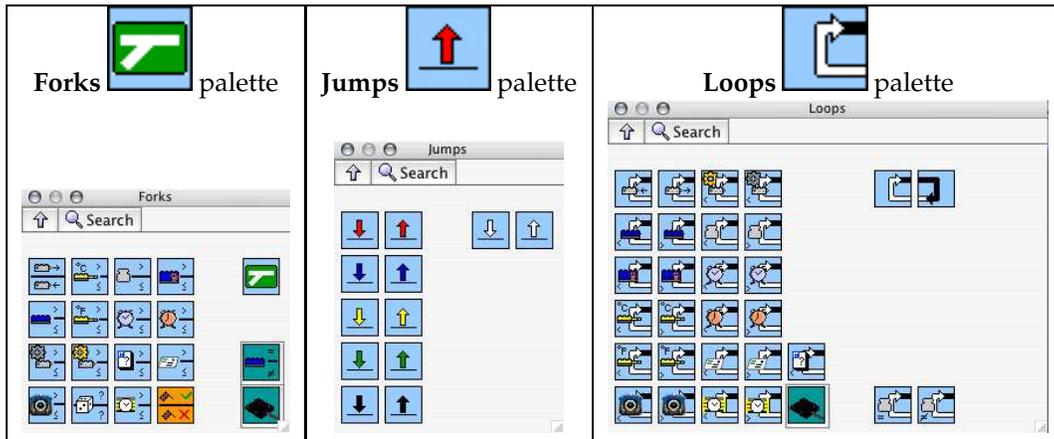
The icons shown below are included on the **Forks** palette:

touch sensor fork	light sensor fork	fork merge
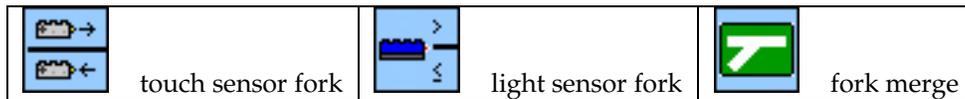
The icons shown below are included on the **Jumps** palette:

black land	black jump	blue land	blue jump
green land	green jump	red land	red jump
yellow land	yellow jump		
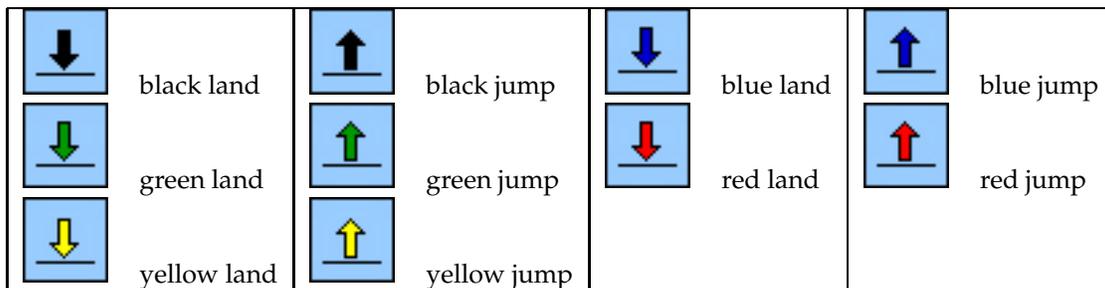
The icons shown below are included on the **Loops** palette:

start of loop		end of loop	
loop while touch sensor is pushed		loop while touch sensor is released	
loop while light sensor value is greater than		loop while light sensor value is less than	

4