

lab A.3: introduction to RoboLab

Name:

vocabulary

- RoboLab
- communication tower
- canvas
- icon
- drag-and-drop
- function palette
- tools palette
- program
- algorithm
- syntax error
- logical error
- wire
- modifier
- numeric constant
- random number

materials

Make sure you have all of the following materials before you start the lab:

- lego robot kit
- communications tower

instructions

1. start up RoboLab

- Find the **RoboLab** icon on your computer and double-click on it to start it up.



- Click on **PROGRAMMER**. Double-click on **INVENTOR 4**. Your screen will look like this:



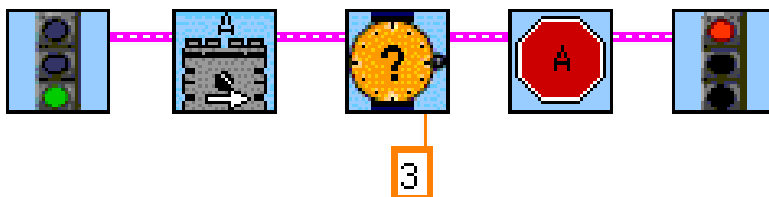
- Now you are ready to start programming in **RoboLab**!

2. find the icons

- Locate the **functions palette**. See if you can find each of the following **icons**. Make a note of which sub-palette each icon belongs to. As you learn what each icon does, write down its **function** in the boxes below.

3. create your first program

- Use the mouse to select, drag and drop RoboLab **icons** onto your canvas and **wire** them together, creating the following program:



- What do you think it does? Write down your ideas below:
-
-

4. download your program



- Find the **communication tower**, which looks like this:
- Connect it to your computer's USB port (ask if you need help finding it).
- Place your robot next to the tower.



- Find the **download arrow** in your RoboLab window that looks like this: It's up near the top, just under "File Edit" menu options. Click on it, and your program will **download** onto your robot!
- If the download works, then the RCX will sing a happy tune :-)
- If you have trouble, check these things:
 1. The RCX must be turned on during the download.
 2. The black part of the RCX should be turned towards your tower.
 3. If there is a lot of light in the room, try covering your RCX and the tower with a piece of paper.
 4. If your download arrow is broken, then you have an error in your program. Make sure that you copied all the icons correctly and that the wires all start and end in the right places.
 5. Make sure the **touch sensor** is connected to **port 1**.

5. run the program

- After the program downloads successfully, put your robot on the floor and turn it on.
 - Then press **RUN**.
 - What did your robot do? Is that what you expected? Write your answer below.
-
-

6. modify the program

- Now change the program to make your robot go in a straight line for 5 seconds and then stop. **Edit** the program in RoboLab, **download** it again and **test** it. Repeat this process until it works!
- Then in the box below, draw your code:



7. programming challenges

- Complete as many of the following programs as you can.
- After you get each program to work, draw the code in the boxes provided on the next page.

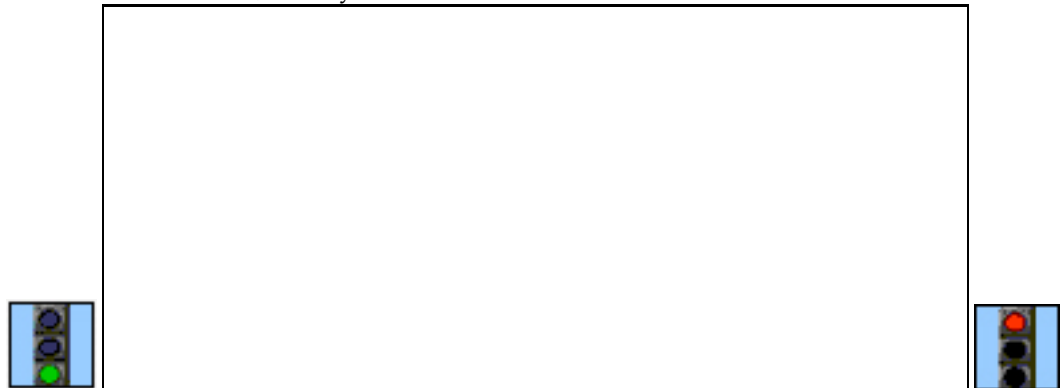
(a) Program the robot to go forward for 2 seconds and then go backward for 2 seconds and then stop. Is it back where it started from?



(b) Program the robot to go in a square.


i. Design your code before you start programming. Write your *algorithm* below.


ii. Now in the box below, draw your **code**




8. random numbers

- The next challenges use the concept of **random numbers**. There are two new icons that you will

need for these challenges: the *wait for random time* icon, which looks like this:  and the

motor random icon, which looks like this: .

-  **Here is how the *wait for random time* icon works:**

The icon tells the robot to wait for a random amount of time. The default is to wait for a random

amount of time between 0 and 5 seconds. You can use the **numeric constant** modifier to set the maximum random time (in seconds). For instance, we can set the maximum random time to 10 seconds in order to pick a random time between 0 and 10 seconds. You should set the modifier to the maximum random number you want to wait for.



• **Here is how the *motor random* icon works:**

The icon turns on motors to run in a randomly chosen direction (either forward or in reverse). You can use the port modifier to choose which motors to turn on by stringing together any combination of output port modifiers A, B and C. You can also choose the power level by stringing a single modifier indicating the power level for the motors.



9. programming challenges

- Complete as many of the following programs as you can.
- After you get each program to work, draw the code in the boxes provided on the next page.

(a) Program the robot to go forward for a **random** number of seconds (between 0 and 7), and go in a random direction for 6 seconds.



(b) Program the robot to play a song using the music notes.



(c) Program the robot to go forward for 2 seconds using power level 1 while playing note A; then go forward for 2 seconds using power level 2 while playing note B; then go forward for 2 seconds using power level 3 while playing note C; then go forward for 2 seconds using power level 4 while playing note D; then go forward for 2 seconds using power level 5 while playing note E; then stop.

