

A PharoThings Tutorial

Alex Oliveira

November 6, 2018

Copyright 2017 by Alex Oliveira.

The contents of this book are protected under the Creative Commons Attribution-ShareAlike 3.0 Unported license.

You are **free**:

- to **Share**: to copy, distribute and transmit the work,
- to **Remix**: to adapt the work,

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page:

<http://creativecommons.org/licenses/by-sa/3.0/>

Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.



Your fair dealing and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license):

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Contents

Illustrations	ii
1 Lesson 3 - LED Flowing Lights	1
1.1 What we need?	1
1.2 Experimental theory	1
1.3 Experimental code	2
1.4 In the next lesson	2

Illustrations



Lesson 3 - LED Flowing Lights

Now we can play with the LEDs, turn them on, off, and blink. Let's put 8 LEDs on the breadboard and create a code to turn on/off one at a time. As we did in the last lesson, let's write the first code in playground and then create a class with methods to better control the flow of LED lights.

1.1 What we need?

We are using the lesson 1 setup, but with 8 LEDs and 8 resistors.

Components

- 1 Raspberry Pi connected to your network (wired or wireless)
- 1 Breadboard
- 8 LEDs
- 8 Resistors 330ohms
- Jumper wires

1.2 Experimental theory

We saw in the lesson 1 how to connect the LED and resistors on breadboard. Now we will do the same, but putting more 7 LEDs and resistors in the breadboard.

...

Connecting remotely

Through your local Pharo image, let's connect in the Pharo image by running on Raspberry, enable the auto-refresh feature of the inspector, and open the inspector.

Run this code in your local playground:

```
remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: #[193 51 236
    212] port: 40423)
GTInspector enableStepRefresh.
remoteBoard := remotePharo evaluate: [ RpiBoardBRev1 current].
remoteBoard inspect.
```

1.3 Experimental code

In your inspect window (Inspector on a PotRemoteBoard), let's initialize the led and set the pin 7 to be in digital output mode as we did in the last lesson:

```
led := gpio7.
led beDigitalOutput.
```

To blink the LED let's create a simple loop to change the value of the LED every 1 second by 10 times. To change the value of the object (led value), let's call the method `toggleDigitalValue`, as we saw previously:

```
[ 10 timesRepeat: [
    led toggleDigitalValue.
    (Delay forSeconds: 1) wait
] ] forkNamed: 'BlinkerProcess'.
```

Run this code, as shown in Figure *and... cool! Now your LED is blinking!

Change the values to repeat more times and to wait less time between toggling. This will cause the LED to blink faster.

1.4 In the next lesson

In this tutorial, you learned how to blink a LED by typing some code in the remote inspector. But with Pharo we can do more! Pharo is a new generation reflective language and programming environment. And in the next lesson, let's start using object-oriented programming, OOP. Let's create a simple application, writing classes and methods, all remotely.