A PharoThings Tutorial

Allex Oliveira

November 28, 2018

Copyright 2017 by Allex 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

Layout and typography based on the `sbabook` LaTeX class by Damien Pollet.

Contents

Illustrations ii

1 Lesson 2 – Blinking LED 1

1.1 What do we need? 1

1.2 Experimental code 2

1.3 In the next lesson 2

Illustrations

1-1 Remote playground. 2

CHAPTER 1

Lesson 2 – Blinking LED

Now we can play with the LEDs, turn them on and off. Let's use this basic setup to write some code on the inspector playground to blink the LED. Next, we will learn how to remotely create a very simple application using classes, methods, and instances to control the LED.

1.1 What do we need?

We are using the same setup as the last section: 1 Raspberry Pi, 1 Breadboard, 1 LED, 1 Resistor 330ohms. If you didn't do the last lesson to understand how to do the connections, go back to Chapter 2 and do it.

Connecting remotely

Through your local Pharo image, let's connect to 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.
```

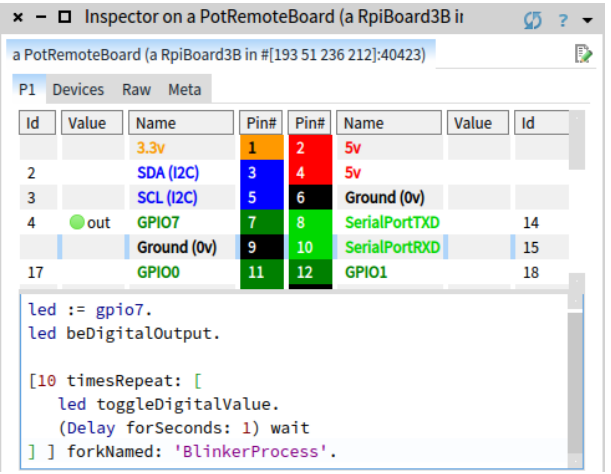

Figure 1-1 Remote playground.

1.2 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 1-1 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.

1.3 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! And in the next lesson, let's start to use OOP (object-oriented programming). Let's create a simple application, write classes and methods, all remotely.