A PharoThings Tutorial

Allex Oliveira

November 7, 2018

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

Contents

Illustrations

CHAPTER 1

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 let's use 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 procedure

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

- Connect the Ground PIN from Raspberry in the breadboard blue rail (-).

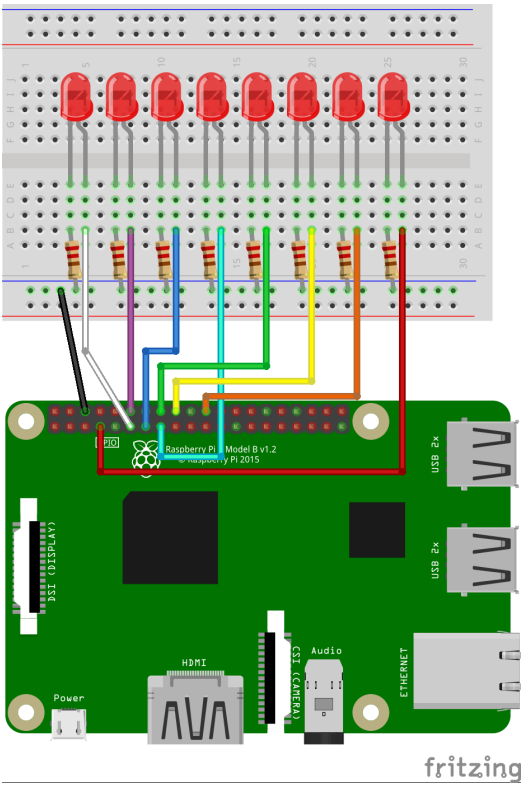

Figure 1-1 Physical connection 8 LEDs.

- Then connect the 8 resistors from the blue rail (-) to a column on the breadboard, as shown below;
- Now push the LED legs into the breadboard, with the long leg (with the kink) on the right;
- And insert the jumper wires connecting the rigth column of each LED to GPIO from 0 to 7, as shown in the Picture 1-1.

The Figure 1-1 shows how the electric connection is made:

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 8 LEDs and set the GPIOs to be in digital output mode:

```
code...
```

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:

```
code...
```