Live Programming IoT devices
With Pharo Things

Do Hoang

University of Science And Technology of Hanoi

June 15, 2019

1 Lesson 4 - LED Flowing Lights

Section 1

Lesson 4 - LED Flowing Lights

What do we need

Components

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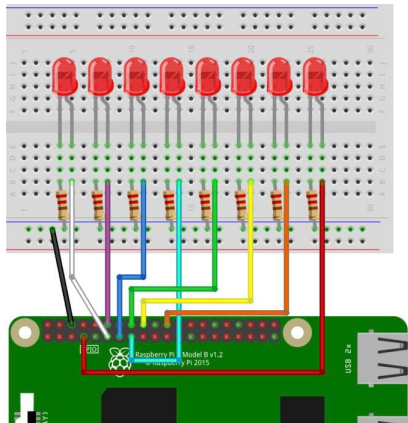- Jumper wires

Schema connection 8 LEDs

Figure 1: Schema connection 8 LEDs
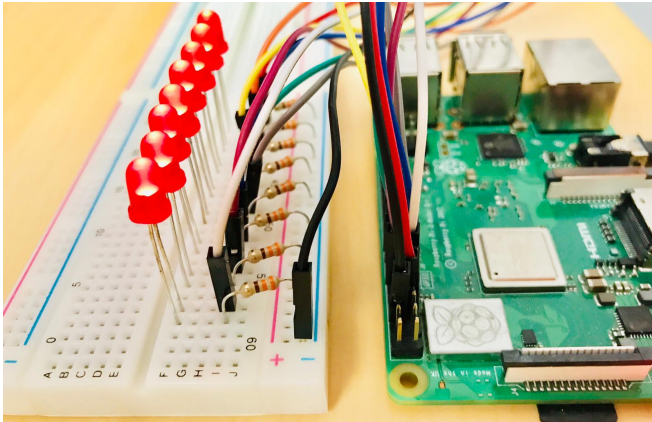
Physical connection 8 LEDs

Figure 2: Physical connection 8 LEDs

Experimental process

- Connect the Ground PIN from Raspberry in the breadboard blue rail (-).
- Then connect the 8 resistors from the blue rail (-) to a column on the breadboard.
- 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 right column of each LED to GPIO from 0 to 7.

Connecting remotely

```
remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip:
    #[193 51 236 212] port: 40423)
GTInspector enableStepRefresh.
remoteBoard := remotePharo evaluate: [ RpiBoard3B current].
    remoteBoard inspect.
```

Experimental code

- Let's create an array and initialize the 8 LEDs, putting each one in a position of the array.

```
gpioArray := { gpio0. gpio1. gpio2. gpio3.
    gpio4. gpio5. gpio6. gpio7 }.
gpioArray do: [ :item | item beDigitalOutput ].
```

Experimental code

- In the previous lesson, we use `toggleDigitalValue` to change the value of the object (Led value)

```
gpioArray do: [ :item | item toggleDigitalValue ].
```

Experimental code

- Let's put a Delay after changing the led value, to wait a bit time before to changethenextLEDvalue. Let's also put this inside a process using the method forkNamed:

```
[

    gpioArray do: [ :item | item toggleDigitalValue.
    (Delay forSeconds: 0.3) wait ].
] forkNamed: 'FlowingProcess'.
```

Result

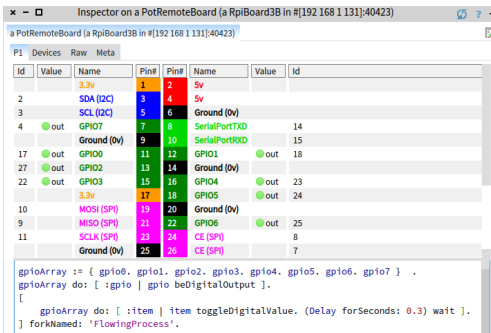- Execute this code and... cool! Now your LEDs are on by flowing an ordering!
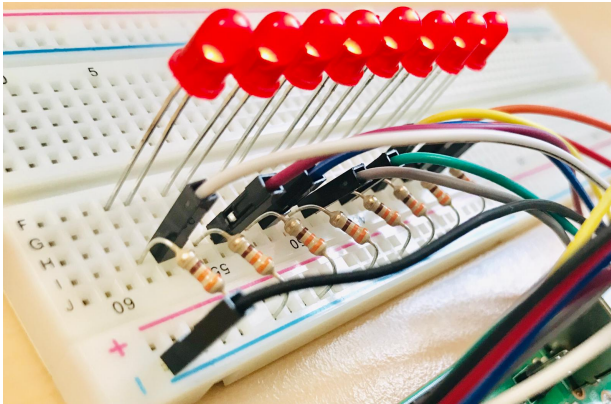

Figure 3: Code on Inspector

Result

Figure 4: LEDs turn On.

Adding features

Control loop time?

```
[ 2 timesRepeat: [
    gpioArray do: [ :item | item toggleDigitalValue.
    (Delay forSeconds: 0.1) wait ].
] ] forkNamed: 'FlowingProcess'.
```

Adding features

Reversing the flow

```
[ 2 timesRepeat: [
  gpioArray reverseDo: [ :item | item toggleDigitalValue.
  (Delay forSeconds: 0.1) wait ].
] ] forkNamed: 'FlowingProcess'.
```
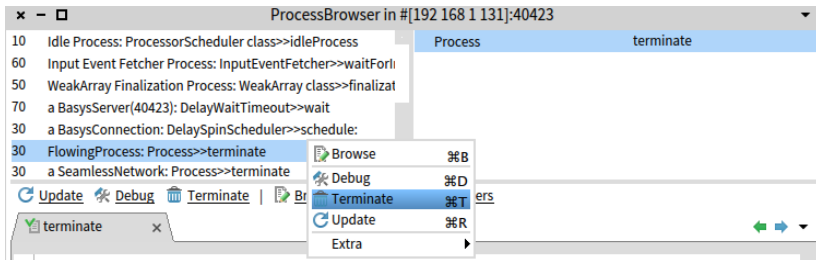
Terminating the process

- Call the **Remote Process Browser**

Figure 5: Remote Process Browser