

# A PharoThings Tutorial

Alex Oliveira

January 11, 2019

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

<b>Illustrations</b>	<b>ii</b>
<b>1 Installations</b>	<b>1</b>
1.1 Installing PharoThings on Raspberry Pi . . . . .	1
1.2 Execute PharoThings on Raspberry . . . . .	4
1.3 Connecting Pharo client on remote Pharo . . . . .	5
1.4 In the next chapter . . . . .	7

# Illustrations

1-1	Installing PharoLauncher. . . . .	2
1-2	Download Pharo 61. . . . .	2
1-3	Open your Pharo image. . . . .	3
1-4	Loading PharoThings. . . . .	3
1-5	Copying PharoARM. . . . .	4
1-6	Copying the folder on your Raspberry. . . . .	4
1-7	Server up and running. . . . .	5
1-8	Remote GPIO inspector . . . . .	6

# Installations

The first step you need to do to get started with PharoThings is to install an Operating System in your Raspberry Pi. When you buy a Raspberry Pi, the OS is not factory installed.

Raspbian is the Foundation's official supported operating system, a Linux OS based on Debian Stretch to run in ARM processors.

Once you've installed the Raspbian, you can install PharoThings and control devices remotely.

## 1.1 Installing PharoThings on Raspberry Pi

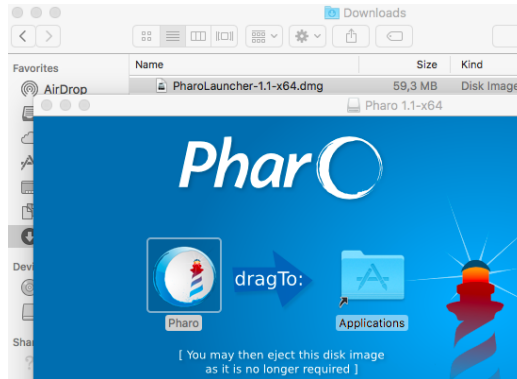
Install PharoThings requires to get Pharo, PharoThings and an ARM virtual machine. We will build the image on our local computer with the correct files and after we will copy to a Raspberry Pi and execute it there. In the third step, we will connect the Pharo from the local computer into Pharo of Raspberry Pi.

### Download PharoLauncher

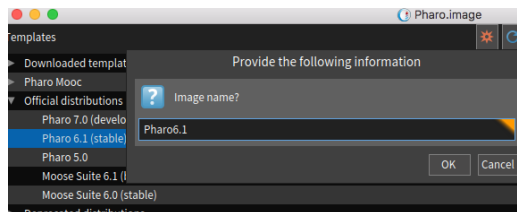
Use the PharoLauncher (an application to help you running multiple versions and images of Pharo) and install Pharo 6.1. You can get the launcher from <http://pharo.org/download>. You can also directly install a version of Pharo from the same place.

### Download Pharo 61

Run the Pharo Launcher. Double click the distribution you want to create a image and give a name to image (see Figure 1-2). A short name and without



**Figure 1-1** Installing PharoLauncher.



**Figure 1-2** Download Pharo 61.

spaces is recommended, because we will type this name and path in command line on Linux.

## Execute your Pharo image

Launch the image as shown in Figure 1-3. A folder with the image name will be created inside the folder Pharo: `/Users/your_user_name/Documents/Pharo/`

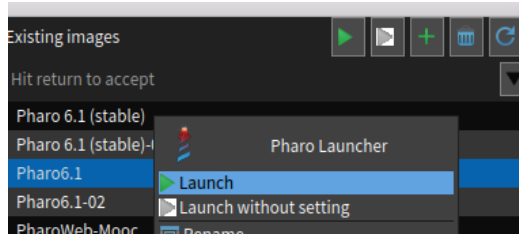
In this example the folder is `/Users/my_user_name/Documents/Pharo/Pharo6.1`

## Load PharoThings

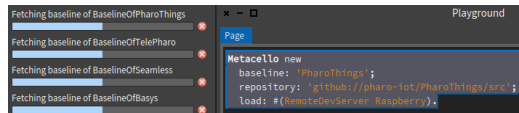
Open Playground and execute this command to install the server part of PharoThings (as shown in Figure 1-4):

```
Metacello new
  baseline: 'PharoThings';
  repository: 'github://pharo-iot/PharoThings/src';
  load: #(RemoteDevServer Raspberry).
```

## 1.1 Installing PharoThings on Raspberry Pi



**Figure 1-3** Open your Pharo image.



**Figure 1-4** Loading PharoThings.

Then configure image to disable slow browser plugins (instead remote browser will be much slower):

```
[ClySystemEnvironmentPlugin disableSlowPlugins.
```

## Snapshot your Image

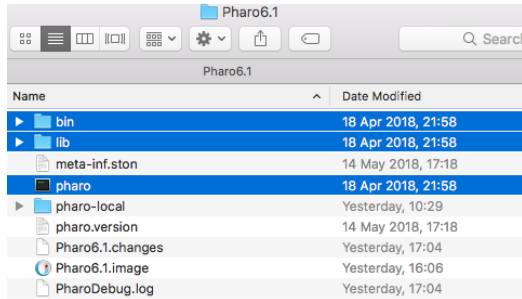
In Pharo, click and “Save and Quit”. This way all your code and configurations are saved and ready to be reused.

## Download the VM

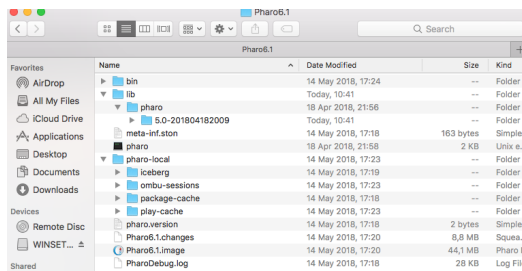
- Download ArmVM from <http://files.pharo.org/vm/pharo-spur32/linux/armv6/latest.zip>.
- Unzip it
- Copy the files shown in Figure 1-5 to Pharo folder `/Users/your_user_name/Documents/Pharo/pharo_image_folder`

## Copying Sources

Copy the file `PharoV60.sources` from the folder `/Applications/Pharo.app/Contents/MacOS` to folder `/Users/your_user_name/Documents/Pharo/images/pharo_image_folder/lib/pharo/5.0-201804182009/`



**Figure 1-5** Copying PharoARM.



**Figure 1-6** Copying the folder on your Raspberry.

## Copy to the Raspberry

Copy this folder to your Raspberry Pi (via flashdrive, network etc). The folder must have the structure shown in Figure 1-6.

## 1.2 Execute PharoThings on Raspberry

### Turn on your Raspberry and connect it to the network.

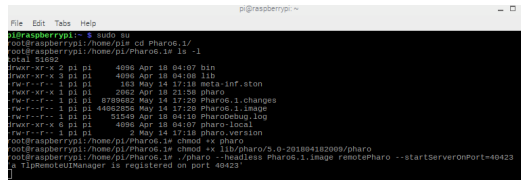
In this example, the folder Pharo6.1 was copied to folder /home/pi/.

Is necessary apply execute permissions on the Pharo files, using the command `chmod +x`

```
chmod +x /home/pi/Pharo6.1/pharo
chmod +x /home/pi/Pharo6.1/lib/pharo/5.0-201804182009/pharo
```



### 1.3 Connecting Pharo client on remote Pharo



```
pi@raspberrypi: ~  
File Edit Tabs Help  
root@raspberrypi:~# cd /home/pi/pharos.1/  
root@raspberrypi:/home/pi/pharos.1# ls -l  
total 5180  
drwxr-xr-x 2 pi pi 4096 Apr 18 04:07 bin  
drwxr-xr-x 3 pi pi 4096 Apr 18 04:08 lib  
rw-r--r-- 1 pi pi 163 May 14 17:18 meta-inf.ston  
drwxr-xr-x 1 pi pi 2082 Apr 18 21:28 Pharo  
rw-r--r-- 1 pi pi 8789682 May 14 17:20 Pharo.1.changes  
rw-r--r-- 1 pi pi 4888205 May 14 17:20 Pharo.1.image  
rw-r--r-- 1 pi pi 51549 Apr 18 04:10 PharoDebug.log  
drwxr-xr-x 4 pi pi 4096 Apr 18 04:07 pharo.local  
rw-r--r-- 1 pi pi 2 May 14 17:18 pharo.version  
root@raspberrypi:/home/pi/pharos.1# chmod +x lib/pharo/5.0-201804182009/pharo  
root@raspberrypi:/home/pi/pharos.1# pharo --headless pharo.1.image remotePharo --startServerOnPort=40423  
TlpRemoteUIManager is registered on port 40423
```

Figure 1-7 Server up and running.

#### Start Server

Start the Pharo typing the following command in the Terminal on your Raspberry:

```
[ pharo --headless Server.image remotePharo --startServerOnPort=40423
```

If all is right, you will see the answer:

```
[ 'a TlpRemoteUIManager is registered on port 40423'
```

So now we have the Raspberry running the TelePharo on TCP port 40423 (as shown in Figure 1-7) and we can connect into it from another computer.

### 1.3 Connecting Pharo client on remote Pharo

Open again the Pharo on your local computer and execute this command to install the PharoThings client:

```
[ Metacello new  
baseline: 'PharoThings';  
repository: 'github://pharo-iot/PharoThings/src';  
load: 'RemoteDev'.
```

Type this command to connect to the remote TelePharo on Raspberry (change the IP to your Raspberry IP):

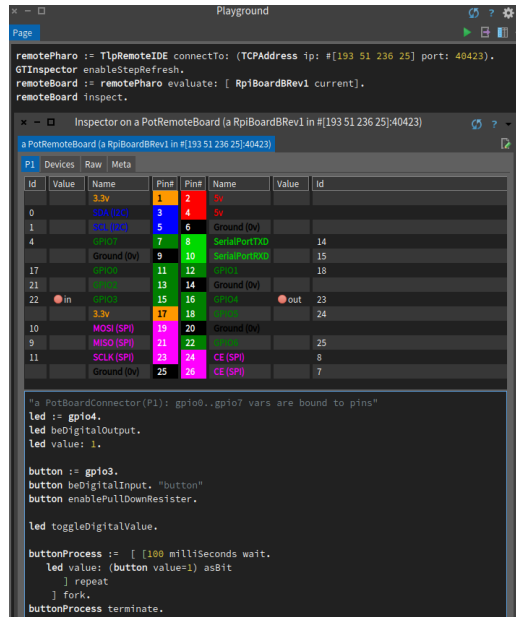
```
[ remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: #[193 51 236  
167] port: 40423)
```

Here we are using specialized Raspberry tools. They require the auto-refresh feature of inspector which is not enabled by default in Pharo 6. To activate it evaluate:

```
[ GTInspector enableStepRefresh
```

So for your board model you need to choose an appropriate board class. For Raspberry, it will be one of the RpiBoard subclasses. Currently, you can use the following classes according to the models:

- RpiBoardBRev1: Raspberry Pi Model B Revision 1



**Figure 1-8** Remote GPIO inspector

- RpiBoardBRev2: Raspberry Pi Model B Revision 2
- RpiBoard3B: Raspberry Pi Model B+, Pi2 Model B, Pi3 Model B, Pi3 Model B+

With the chosen class evaluate the following code to open an inspector:

```
[remoteBoard := remotePharo evaluate: [ RpiBoard3B current].
remoteBoard inspect.
```

And will be open the inspector showing the PIN scheme (as shown in Figure 1-8)

## GPIOs

The board inspector shows a layout of pins similar to Raspberry Pi docs. But here it is a live tool which represents the current pins state.

In the picture the board is shown with two configured pins: gpio3 and gpio4, which are connected to physical button and led accordingly.

Digital pins are shown with green/red icons which represent high/low (1/0) values. In case of output pins you are able to click on the icon to toggle the value. Icons are updated according to pin value changes. If you click on physical button on your board the inspector will show the updated pin state by changing its icon color.

#### 1.4 In the next chapter

The evaluation pane in the bottom of the inspector provides bindings to gpio pins which you can script by `#doIt/printIt` commands. The example shows expressions which were used to configure a button and led.

#### 1.4 In the next chapter

Now that we have installed the Operation System and PharoThings in the Raspberry Pi, we can play with LEDs, sensors, LCD displays and more.

In the next chapter we will see how turning on/off a LED using PharoThings.

