

A PharoThings Tutorial

Alex Oliveira

November 29, 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 Installations	1
1.1 Installing OS on RASPBERRY (RASPBIAN)	1
1.2 Copying Raspbian files on MAC OSX	2
1.3 Copying Raspbian files on Linux	2
1.4 Copying Raspbian files on Windows	2
1.5 Installing the Raspbian in Raspberry Pi	2
1.6 Installing PharoThings on Raspberry Pi	3
1.7 Execute PharoThings on Raspberry	6
1.8 Connecting Pharo client on remote Pharo	7
1.9 In the next chapter	8

Illustrations

1-1	Preparing SD Card.	2
1-2	Copying NOOBS.	2
1-3	Installing Raspbian.	3
1-4	Installing PharoLauncher.	3
1-5	Download Pharo 61.	4
1-6	Open your Pharo image.	4
1-7	Loading PharoThings.	5
1-8	Copying PharoARM.	5
1-9	Copying the folder on your Raspberry.	6
1-10	Server up and running.	7
1-11	Remote GPIO inspector	8

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.

1.1 Installing OS on RASPBERRY (RASPBIAN)

In this chapter, we will download and install NOOBS (New Out Of the Box Software). NOOBS is an easy operating system installer which contains Raspbian (<https://www.raspberrypi.org/downloads/raspbian/>) and LibreELEC (<https://libreelec.tv>).

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

Download

You can download an official image from the Raspberry Pi website Noobs downloads page (<https://www.raspberrypi.org/downloads/noobs/>). You will download a zip file and extract the files to your SD card.

Copying

You will need a computer with an SD card reader to install the image. This process basically extracts the files from the zip file downloaded into an SD card formatted and start the Raspberry Pi with this SD card.

You can go directly to your operating system by clicking on the links below:

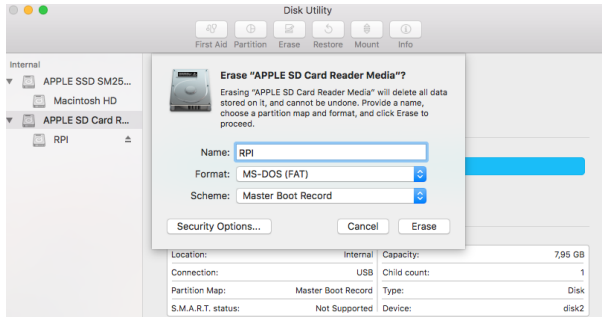


Figure 1-1 Preparing SD Card.

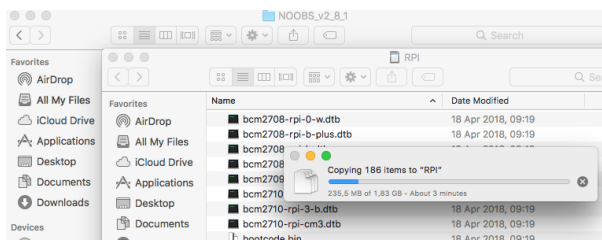


Figure 1-2 Copying NOOBS.

1.2 Copying Raspbian files on MAC OSX

- Open “disk utility”, select the SD Card and Erase (Format MS-DOS FAT) as shown in Figure 1-1.
- Copy the files from folder NOOBS_xxx to SD Card as shown in Figure 1-2.

1.3 Copying Raspbian files on Linux

1.4 Copying Raspbian files on Windows

1.5 Installing the Raspbian in Raspberry Pi

Insert the SD Card on Raspberry and turn it on. Select “Raspbian” and “Yes” as shown by Figure 1-3.

In a few minutes, you will have your Raspberry Pi running Raspbian OS. Now you can install PharoThings and control devices remotely.

1.6 Installing PharoThings on Raspberry Pi

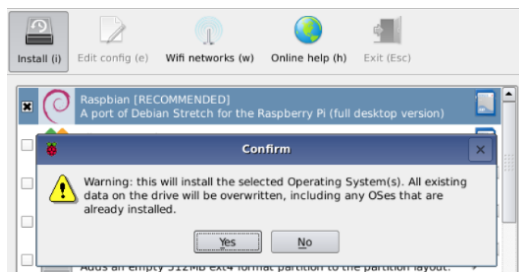


Figure 1-3 Installing Raspbian.

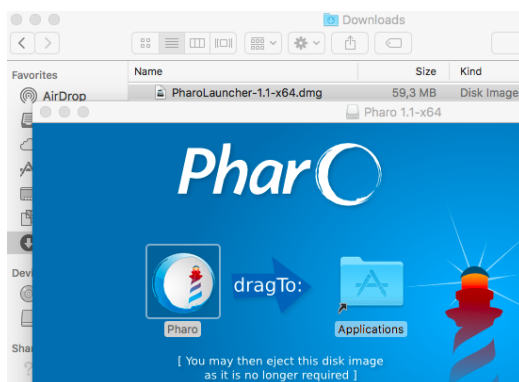


Figure 1-4 Installing PharoLauncher.

1.6 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.

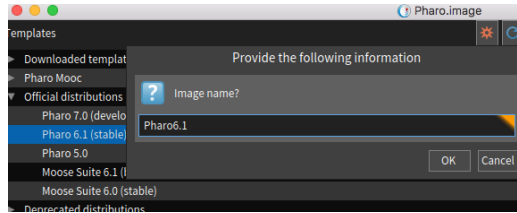


Figure 1-5 Download Pharo 61.

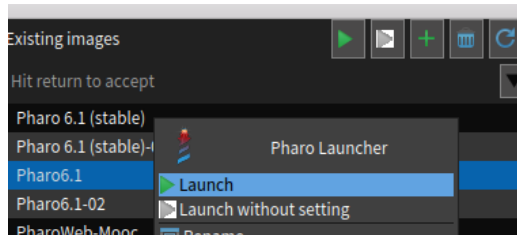


Figure 1-6 Open your Pharo image.

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 1-5). A short name and without 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 1-6. 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 1-7):

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


1.6 Installing PharoThings on Raspberry Pi

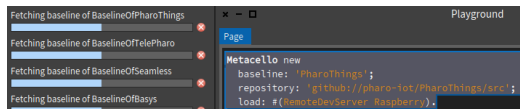


Figure 1-7 Loading PharoThings.

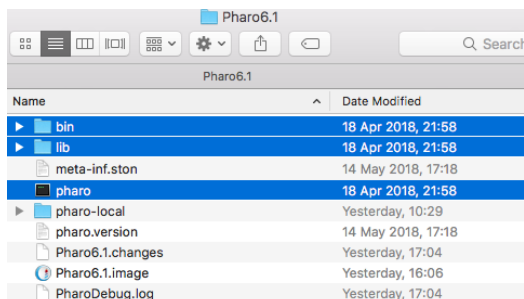


Figure 1-8 Copying PharoARM.

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 1-8 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/`

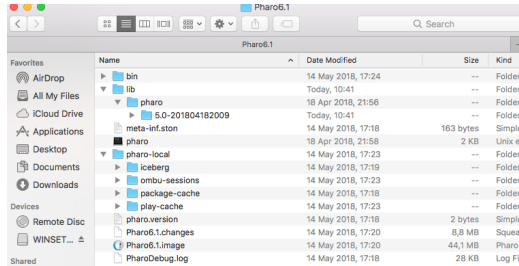


Figure 1-9 Copying the folder on your Raspberry.

Copy to the Raspberry

Copy this folder to your Raspberry Pi (via flashdrive, network etc). The folder must have the structure shown in Figure 1-9.

1.7 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
```

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 1-10) and we can connect into it from another computer.

1.8 Connecting Pharo client on remote Pharo

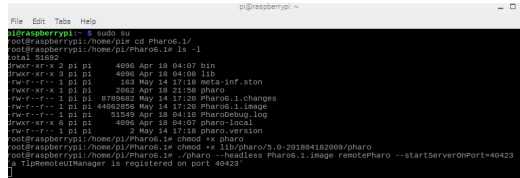
A terminal window titled 'pi@raspberrypi ~' showing the execution of a Pharo script. The script starts with 'cd /home/pi/Pharo' and 'ls -l', showing a directory listing of files like 'bin', 'meta-inf', 'Pharo', 'PharoChanges', 'PharoImage', 'PharoDebugLog', and 'PharoLocal'. It then runs 'chmod +x lib/pharo/5.0-201804120009/pharo' and 'startServerOnPort=40423'. The final output is 'TipRemoteUIManager is registered on port 40423'.

Figure 1-10 Server up and running.

1.8 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
- 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: [ RpiBoardBRev1 current].
remoteBoard inspect.
```

And will be open the inspector showing the PIN scheme (as shown in Figure 1-11)

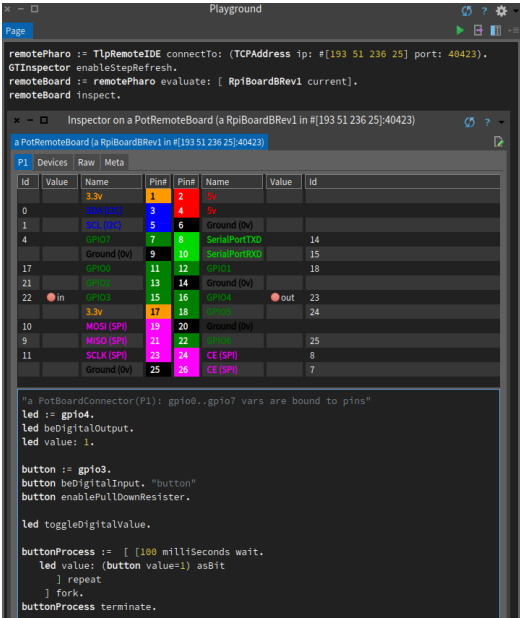


Figure 1-11 Remote GPIO inspector

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.

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.

1.9 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 turing on/off a LED using PharoThings.